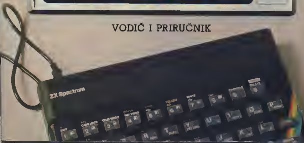
KOMPJUTOR U KUĆI

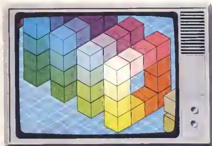
PETER LAURIE

VODIČ I PRIRUČNIK



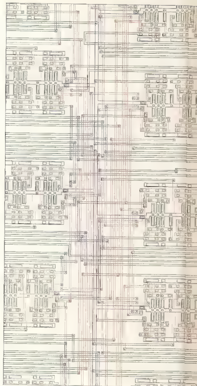
KOMPJUTOR U KUĆI

PETER LAURIE



VODIČ I PRIRUČNIK

CANKARJEVA ZALOŽBA
LJUBLJANA, ZAGREB
1984



Originalni naslov
PETER LAURIE
THE JOY OF COMPUTERS

Prijevod Nikola Tenholer
Suradnik na prijevodu Nenad Jakovina
Programi za Spectrum Siniša Novosel
Uredio i opremio Kazimir Rapoša
Izdala Cankarjeva založba u Ljubljani
Za izdavača Martin Žnideršič

Tisak Tiskarna Ljudske pravice u Ljubljani 1984

SADRŽAJ

PREDGOVOR

Ova knjiga pokušava otvoriti neke nove vidike u jedan novi, čudesni svijet. Svijet koji je još prije nekoliko godina bio rezerviran tek za nekoliko tisuća posvećenih i sjajno plaćenih profesionalaca.

Nagli pad cijena malih kompjutora u posljednje je vrijeme uzrokovao njihovo naglo rasprostiranje diljem svijeta u stotinama tisuća primjeraka. Milijune ljudi najednom muči briga da će ostati kompjutorski nepismeni, što im se čini čak i opasnijim od one klasične nepismenosti. A kada pomisle na vlastitu djecu, brige im se udvostručuju. Mladi kompjutorski stručnjaci koji dolaze ravno sa sveučilišta, postižu visoke zarade radeći kao profesionalci u hardverskoj i softverskoj industriji, a manje naobraženi pojedinci (koji put s iskustvom od samo nekoliko mjeseci na nekom kućnom kompjutoru) postaju sve traženiji u poslovnom svijetu, jer glad za ljudima koji će upravljati manjim kompjutorskim sistemima postaje sve veća.

Moćne kompjutorske korporacije pažljivo podgrijavaju naklapanje o nekoj tajanstvenosti kompjutora i posebnoj vrsti ljudi koji njime upravljaju. Nagađa se o tome kako su kompjutori neki elektronski mozgovi, a programeri vrsta nadljudi. Činjenice, međutim, dokazuju da su kompjutori prilično glupi, a oni koji ih programiraju sasvim obična ljudska bića. Svatko tko uz pomoć prstiju vlastitih ruku zna brojiti od 0 do 7 (treba odmah naglasiti da je to 8 prstiju), može sigurno naučiti programirati. Taj posao uopće nije težak, samo je ponešto zamršen.

Pogrešno je uvjerenje da kompjutor može „misliti" poput čovjeka. On to zaista ne može, jer nije ništa pametniji od obične kosilice za travu. Ali je istina da ga se može vrlo jednostavno natočiti razumom. Treba samo smisliti način i rješenje nekog određenog problema, napisati program, a kompjutor će se tada prilagoditi vašem mišljenju, baciti se na posao i rješavati zadatak kad god se to od njega zatraži i koliko god puta hoćete. Ako tako gledamo, kompjutor bi zajedno sa svojim programom mogao do izvjesne mjere djelovati poput živa bića, jer na svojstven način može konzervirati misli i naume svojih stvaralaca. U struci se često čuje rečenica „ovo računalo zna kada treba uraditi ovo ili ono", kao da se radi upravo o ljudskom biću. A zapravo se htjelo reći da je osoba koja je pisala program za računalo znala kada i što treba učiniti.

Revolucija u obradi podataka prijeti dubokim promjenama u našem načinu života, ali one neće biti ništa teže nagoli druge koje smo postupno asimilirali. U ne tako davnoj prošlosti doživjeli smo revoluciju tiskarstva, industrijsku revoluciju, željezničku, električku, telegrafsku, telefonsku, zrakoplovnu, radio i televizijsku revoluciju. Kompjutorska revolucija samo je još jedan korak u vječnoj težnji ljudske vrste da zavlada svojom okolinom. Čovjek je izmislio strojeve koji će štedjeti i nadilaziti njegove tjelesne snage, a sada je napravio kompjutor koji štedi, a često i nadilazi njegove umne sposobnosti. Bez sumnje, kompjutor je taj koji će nas odvesti u daljine o kojima ni sanjali nismo, kada smo započinjali.

Iako je rad na prvim kompjutorima počeo za vrijeme drugoga svjetskoga rata, što u povijesnim razmjerima i nije tako davno, oni su ubrzo stvorili jednu vlastitu i vrlo bogatu kulturu. Zaista nije moguće razumjeti današnje mikrokompjutore bez ikakva pojma o onome što se ranije događalo, jer je u njih ugrađeno ogromno mnoštvo pretpostavki i ideja koje su se gomilale godinama.

Usprkos povijesti, ritam kojim se nižu promjene je tako brz, da je svatko dobrodošao tko ima neku novu, svježu ideju. Promjene se istodobno događaju na dva fronta. Sam kompjutor postaje svakim danom sve jeftiniji i moćniji. To znači da posao koji je još prije nekoliko godina moglo obavljati samo visoko specijalizirano osoblje na svojim golemim

strojevima danas obavlja sasvim rutinski gomila običnih uredskih službenika. Druga je promjena u tome da je kompjutorska obrada podataka izišla na svjetlo dana i postala svakodnevnim oruđem obična čovjeka. Nisu to više strojevi kojima se može služiti samo bogato nagrađena kasta posvećenih koji govore nekim svojim, samo njima razumljivim jezikom. Oni su se pomiješali s običnim svijetom koga zanima samo gotov posao, a ne zamršeni kompjutorski postupci. To je izmijenilo kompjutore jednako kao što je masovna proizvodnja promijenila lice automobilske industrije.

U početku je automobil bio igračka zanesenjaka. Moglo se, doduše, odvesti od Londona do Beijinga, ali je svatko pri tom morao biti spreman da svoje vozilo usput popravlja nekoliko puta. Čim su se automobili stali prodavati na masovnom tržištu, to se moralo promijeniti. Auto je morao postati pouzdan i udoban. Novom vlasniku ne pada na pamet da svakih dvadesetak milja mijenja klipove u motoru, nesretan je ako se vrata loše zatvaraju ili ako upaljač za cigarete ne radi kako treba. Ista se stvar događa sa kompjutorima. Još prije nekoliko godina tipičan vlasnik kućnog kompjutora bio je čudak srca prepunog sreće kada je svog miljenika mogao tokom noći dva puta rasklopiti na kuhinjskom stolu. Naprotiv, danas deseci tisuća ljudi smatraju da jedino što trebe učiniti kako bi sredili svoje knjigovodstvo, napisali pismo (baš tako, napisali pismo!) ili zaigrali kakvu zanimljivu igru, je kretnja kojom se kompjutor uključuje u električnu mrežu. Ne trebe niti spomenuti da su užasnuti pri pomisli na električno lemilo.

Netko može pomisliti da je svijet kompjutora potpuno, gotovo kirurški sterilan u svojoj hladnoj logici. Sretan sam što mogu ustvrditi da to uopće nije istina. Zapravo, svijet kompjutora je čudesno nalik svijetu visoke mode: prepun je fanatika, šarlatana, luđaka i svakojakih modnih trica — ali i ogromnog broja marljivih, zanimljivih i pametnih ljudi, ponosnih što se nalaze na grudobranu napretka ljudske vrste dajući najbolje od sebe da bi stvari krenule naprijed.

Gledajući s drugog stajališta, svijet kompjutora nalik je američkom divljem zapadu u prošlom stoljeću. Zemlja je tako prostrana, toliko bogata, da nikome jednostavno ne pada na pamet da miruje kao kvočka na jajima: industrija jednostavno usisava ljude u pohlepi za novim snagama i svježim idejama. Draža su joj makar i amaterska dostignuća nego suhe kvalifikacije. Onaj koji je na divljem zapadu znao dobro pucati, a imao je pošteno lice, mogao je odmah postati šerif. Tako i u kompjutorskom svijetu onaj tko zna svoj posao, taj će ga i dobiti. Nitko ne ispituje gdje i kako ga je naučio i kakva titula stoji ispred njegova imena.

Postoje mnoge sile koje vrlo snažno privlače ljude kompjutorima. Na prvom mjestu je, bez sumnje, činjenica da ova industrija daje posao i onda kada je od njega vrlo teško doći. Drugo, ona nudi širom otvoren prostor svakoj vrsti talenta, i treće, financijska dobit je potencijalno velika. Naglo bujanje masovnog tržišta omogućuje zaradu u holivudskom stilu. Ona dva maloljetnika što su napravila prvi Apple kompjutor morala su prodati jedan kalkulator i svoj stari kombi da bi odšla do novca kojim su financirali projekt, a već nakon pet godina bili su milijuneri.

Da li zbog želje za stjecanjem ili iz čista zanosa, no uvijek će se naći ljudi spremnih da slijede svoje zamisli ma kamo ih one vodile. Rezultat je mnoštvo različitih računala, programskih jezika i tehnika pomoću kojih se može gotovo sve izračunati i sve izvesti. I eto, sad najednom toliko toga ima što bi trebalo opisati i objasniti, da bi ova knjiga mogla biti četiri ili pet puta deblja. Ali ja mogu tek otškrinuti prozore što gledaju u ove nove čarobne vrtove i nadati se da će se naći oni koji će smatrati vrijednim da uđu u njih.

PETER LAURIE 1983





MIKROKOMPJUTOR

Danas je većina država shvatila da ekonomska budućnost njihovih zemalja ovisi o kompjutorizaciji iduće generacije radnika. Na slici je grupa djece u jednoj britanskoj školi pri obuci na računalu 480Z kojim ih snabdijeva Ministarstvo za znanost i odgoj



Suprotna strana: kri stupnja u razvoju hardvera.

Gore: kućni osobni ili kompjutor za vježbu na koji se može priključiti i kakav jednostavni printer. Običan televizor služi kao displej, a mali kazetofon kao vanjska memorija za pohranu podataka. Uglavnom ima manje od 64 K memorije.

U sredini: Malo poslovno računalo. Može raditi s 8 ili 16 bitova. Ima vanjsku memoriju na magnetskom disku, matrični ili daisywheel printer i spominjem to kao osobito važno debelu knjigu upute. Može imati između 64 i 128 K memorije i prilagođen je standardnom softveru.

Dolje: U pozadini je nacrtan pravi kompjutor s jednim ili više terminala u prednjem planu. Njime se može istodobno služiti više korisnika. Ispred displeja vidimo tastaturu, lijevo je ploča za digitalno crtanje, desno x y ploter, a u sredini miš — naprava za poseban način unošenja grafičkih i drugih podataka. Sam kompjutor može raditi s nekoliko 8 ili 16 bitnih procesora.

Svaki kompjutor, pa i onaj najmanji, u osnovi se ne razlikuje mnogo od najvećeg IBM-a ili golemog Crays-a koji u svojim ozbiljnim sivim kutijama zaposjedaju bezbrojne prostorije. Prema klasičnoj definiciji „kompjutor je naprava koja služi obradi podataka", ali nam ta spoznanja ne pomaže mnogo da bismo pronikli u srž stvari. Ova bi knjiga trebala poslužiti određenom razumijevanju kompjutora i njegove svrhe, ali onaj koji je željan udžbeničkih definicija neće ih ovdje naći, jer namjera ove knjige i nije da bude udžbenik.

Kompjutori se u osnovi mogu podijeliti u tri glavne grupe: velike kompjutorske sustave, minikompjutore i mikrokompjutore. Razlike se osim onih čisto tehničkih, odnose uglavnom na cijenu i marketing.

Veliki kompjutorski sustav (main-frame) je veliki stroj basnoslovne cijene. Njime upravlja ekipa profesionalaca u posebno za tu svrhu građenim prostorijama, koje mogu stajati koji put i više nego sam kompjutor.

Minikompjutor je plod prvih grčeva pojeftinjenja i minijaturizacije kompjutorske tehnologije. Pojavio se prije kojih desetak godina, a upotrebljava se uglavnom na sveučilištima ili pri većim kompanijama. Ne treba mu posebnog profesionalnog osoblja i namjenski građenih prostorija i dakako mnogo je jeftiniji od velikog sustava, ali još uvijek izvan dohvata čak i bogatijih pojedinaca.

Najvažnija je činjenica da se mikrokompjutorom može služiti doslovce svatko. Ta se grupa može (u ovom času) podijeliti u dvije podgrupe. U prvu spadaju osobni kompjutori, jeftini, ali zato ne previše moćni, a u drugu poslovne računala koja često mogu obavljati mnoge funkcije srednjih pa i velikih sustava.

Ta jednostavna podjela postaje u posljednje vrijeme sve zamršenijom. To je rezultat novih sve većih mogućnosti koje rastu poput lavine uz sve moćnije nove čipove (vidi str. 18-19). Stari 8-bitni mikrokompjutor nema baš prevelikih mogućnosti, ali nova 16-bitna računala gotovo su istih snaga kao minikompjutori, dok je najsvežija generacija mikrokompjutora, kao na primjer 32-bitni Hewlett-Packard, gotovo ravna velikim kompjutorskim sustavima.

Ali sve to i nije previše važno. Ono zanimljivije, a dakako i važnije, ne leži u području hardvera — u samom stroju, već u onom što nazivamo softver, u programu.

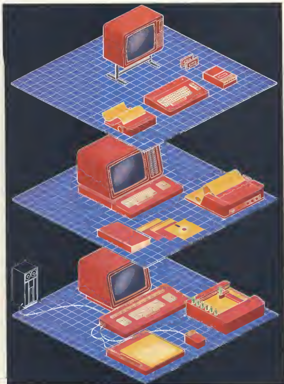
Najbolje ćemo shvatiti mikrokompjutor ako ga usporedimo s jednim dosta kompliciranim električnim pisaćim strojem. Pritisnemo li bilo koju tipku tastature, slovo koje nam se našlo pod prstom pojavit će se odmah na ekranu, ali to 'odmah' predstavlja vrlo zamršen proces koji ćemo razmatrati na slijedećim stranicama. Možemo napisati nekoliko slova ili cijelu riječ, rečenicu, odlomak — pa čak i cijelu knjigu kao što je ova. Sve to možemo i vidjeti na ekranu, i sačuvati na disku ili magnetskoj vrpci. Možemo ponovo vratiti bilo koju stranicu na ekran i načiniti izmjene. Možemo sasvim automatski promijeniti sve 'Cartere' u 'Reagane', ako je u međuvremenu došlo do političkih promjena u američkom vrhu. Možemo načiniti poseban program koji će sastaviti indeks neke knjige. Sve to nije osobito teško. Možda je ponekad zamorno, ali ipak je lakše kada sve ove poslove obavi stroj umjesto čovjeka.

Pretpostavimo da je pred vama neko dosadno zbrajanje. Možda vas jednog dana budu morile brige zbog nadnica koje su iznenada porasle za 5 %, građevni materijal za 30 %, amortizacija pala za 6 %, interesna stopa također za 1 1 %, a vi se upravo spremate osvojiti tržište u Saudijskoj Arabiji. Vaš mikrokompjutor, dakako, sve to može izračunati. Ili ako projektirate most, morate biti sigurni da će svaki nosač biti dovoljno snažan da nosi odgovarajući teret plus teret prometa koji će preko njega teći. Ako se ustanovi da je samo jedan element preslab, a vi ga pojačate, to će povećati teret koji nose drugi, pa sve treba iznova računati.

Ili ako ste nadzorni organ odgovoran za utrošak materijala na nekom gradilištu — koliko opeka treba da bi se podigao zid visok 13, a dugačak 50 metara s otvorima za 16 prozora i četvora vrata? Koliko žbuke treba za taj posao i koliko cementa za temelje? Vaš mikrokompjutor odgovorit će spremno na sva pitanja.

Ili možda uživate u igrama? Na primjer, ludi ste za 'Svemirskim osvajačima' (Space Invaders), što čekate, vaš kompjutor je nestrpljiv da vam i u tome pomogne.

Kompjutoru treba prepustiti sve dosadne zadatke. On će osloboditi svoga gospodara mučnih poslova i omogućiti mu da se bavi kreativnijim poslovima. Ali nije lako natjerati kompjutor na posao. I da budemo sasvim pošteni, na kompjutoru koji radi svoj posao, nema ničeg osobito zabavnoga.



NA ULAZU

Programom smatramo sve ono što upravlja radom kompjutora što prerađuje podatke i daje rezultate Program je i onaj obični a ipak vrlo zakucasti postupak koji se odvija kada na tastaturi pritisnemo tipku s nekim slovom e odmah se na ekranu pojevi slika tog istog slova Pritisak na tipku predstavlja ulazni podatak (data input) a slika slova na ekranu — rezultat (autput)

Na tastaturi kompjutora može se otipkati svako slovo interpunkcijski znak ili neki broj Sve je to kodirano ASCII kodovima (American Standard Code for Information Interchange — Američki standardni kôd za razmjenu informacije Vidi str 185) to jest brojevima koji sežu od 0 do 127 Broj 65 na primjer predstavlja slovo A', razmak između riječi predstavljen je brojem 32 brojka 3 nosi kod 51 a + je 43 i tako dalje

Kompjutor međutim razumija samo jedno prisutnost ili otsutnost električne struje a to se može predstaviti kao da ili ne uključeno ili 'isključeno 1 ili 0 Zbog toga se decimalni brojevi pretvaraju u njihove binarne ekvivalente Nije nužno znati pretvarati decimalne u binarne brojeve treba samo zapamtiti da decimalni broj raste od 0 do devet i da se to događa u jednoj koloni a čim nastavi rasti dalje od 9 formira se nova kolona Binarni brojevi naprotiv rastu samo od 0 do 1 prije nego što će formirati novu kolonu Stoga se binarno 2 predstavlja kao 10, četiri kao 100 osam je 1000 a šesnaest 10 000 U svijetu kompjutora binarni brojevi nazivaju se bitovima a često se upotrebljava kratica b (malo b)

A sada da se vratimo izvođenju onog jednostavnog programa kojim se slovo 'A prikazuje na ekranu Pritisnuvši tipku 'A' tastatura je kompjutoru poslala signal 65 (binarno 01000001) ovaj je odmah potražio u vlastitoj memoriji (ROM — Read Only Memory vidi str 24-25) oblik koji odgovara kôdu i naslikao sliku slova A na ekranu

Ako ste otipkali 3 + 4 tastatura šalje kôdove 51 43 52 a kompjutor odmah na ekranu prikazuje oblike 3' '+' i 4' Nije teško odgonetnuti da 3 plus 4 daje sedam Ali ako zatražite da stroj to učini trebat će za to jedan dosta složen program Programski jezik BASIC sadrži već takav program koji zna (ili bolje reći zna programer koji je sastavio program) kako i što treba učiniti sa zadanim brojevima

Ako pogledate tabelu na stranici 185 vidjet ćete da se ova shema ne uklapa baš glatko u umnoške broja 10 Mnogo bolje pristaju umnošci broja 16, i to je ono što nazivamo heksadecimalnim brojčanim sustavom' Brojeći heksadecimalno ređaju se brojevi na slijedeći način 1 2, 3 4 5 6 7 8 9 A B, C D E F Treba zapamtiti da je za kompjutor broj 16 neobično važan kao uostalom i broj 8 Osam bita predstavlja jedan bajt' (byte upotrebljava se kratica B') Kapacitet kompjutorove unutarnje memorije a isto tako i vanjske (disk ili vrpca) označava se i mjeri upravo bajtovima

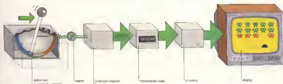
Budući da se jedan bajt sastoji od osam bitova on može predstavljati bilo koju vrijednost od dva na osmu potenciju Dva na osmu iznosi 256 pa se tako jednim bajtom može kodirati 256 različitih stvari Ali sistem ASCII kodiranje zaposjeda mjesta od 0 do 127, dakle radi samo sa 7 bita ili s polovicom od 256 mogućnosti (Ako ovo izgleda previše zamršeno treba zapamtiti da 1 bit pokriva područje od 0 do 1 2 bita od 0 do 3 4 bita od 0 do sedam 7 bita od 0 do 127 a 8 bitova od 0 do 256 ili 256 mogućnosti)

Ostatak od 128 mogućih kôdova upotrijebljen je na dva različita načina Prvi se primjenjuje onda kad se ASCII kodovi šalju nekim dugačkim vodom na primjer telefonskom linijom Tada uvijek postoji mogućnost pogreške u prijenosu U svakom poslanom bajtu osmi bit se postavlja na položaj 1 ili 0 već prema tome da li se radi o parnom ili neparnom broju jedinica u ostatku bajta Na drugom kraju kompjutor koji prima poruku provjerava da li je to točno Ako je u nekom bajtu nešto krivo osmi bit također će biti krivo postavljen i kompjutor-primalac odmah će zatražiti da se poruka ponovi To se zove kontrola jednakosti (parity checking) pa se stoga osmi bit i naziva paritetni bit'

Drugi način korištenja osmog bita ili gornje polovice od 127 ASCII znakova obavlja se u samom kompjutoru gdje ne postoji mogućnost greške u prijenosu Taj je prostor iskorišten za različite grafičke znakove jednake veličine kao i slova Ti se grafički znakovi mogu koristiti i za sastavljanje jednostavnih slika na ekranu

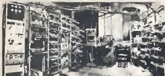
Dolje Riječ JOY pretvorena u ASCII znakove Svaki od 3 bajta što se sastoji od 8 bitova može se prikazati kao niz prekidača koji stoje u položajima 1 i 0'

Kako radi analogno digitalni konvertor napon kojega treba izmjeriti predstavljen je horizontalnom crtom na crtežu Konvertor generira napon za komparirane koji raste u zadanim intervalima pa će prije ili kasnije doseći visinu napona kojega treba izmjeriti U času kada se naponi izjednače na izlazu konvertora javlja se signal Zakašnjenje signala ravno je visini mjerenog napona



Palica (joystick) upotrebljava se u kompjutorskim igrama. Njome kompjutor prima ulaze (input) i prevodi ih u izlaze (output). Pomicanjem ručice mijenja se vrijednost promjenljivog otpornika. Različit napon što dolazi iz njega srazmjeran je položaju ručice. Jedan analogno digitalni konverter (pretvarač) prevodi te različite napone u brojčane vrijednosti. Brojevi se tada upisuju u memorijske ćelije kompjutora, a procesor pomoću njih izračunava položaje svemirskih osvajača na ekranu.

TISKANA PLOČA





Prvi pravi kompjutor u svijetu MARK 1 Sagrađen u Engleskoj na Sveučilištu u Manchesteru koncem četrdesetih godina Nekoliko tisuća elektronskih cijevi u njemu mogio je koji put raditi gotovo 4 sata bez kvara Bio je programiran na osnovi 32 broja upisivanih natraške Alan Turing (vidi str 162—5) glavni programer nikako nije vidio razloga zašto bi se kompjutor prilagođavao korisnicima i njihovoj nesposobnosti da misle poput njega Kao što se vidi uzajamna naklonost između programera i korisnika već je tada bila kamen spoticanja Današnji ekvivalent Marka 1 ručni salkalkulator po prilici je jednakih mogućnosti kao ovaj prvi kompjutor

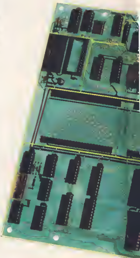
Ako izvadite kompjutor iz njegova kućišta vidjet ćete bez obzira na to radi li se o golemom Cray-u ili najmanjem Sinclairu nešto sasvim nalik slici s desne strane Dugoljasta crna tijela su integrirani sklopovi, integrirci ili čipovi Izvana se može vidjeti da su dugački samo koji centimetar s nizom metalnih nožica (pinova) sa svake strane Nazvani su integriranim sklopovima, jer utjelovljuju u jednom komadu gomilu tranzistora otpornika, kondenzatora i ostalih aktivnih i pasivnih elektronskih elemenata Utaljeni su u plastično kućište koje je zapravo mnogo veće od samog čipa Da nije tako bilo bi nespretno rukovati ovim sitnim i krhkim stvarcama

Ima na stotine čipova koji mogu obavljati isto toliko različitih poslova Neki su predviđeni za obavljanje logičkih funkcija (vidi str 20-21) drugim je određeno da iz mlaza podataka što struje kroz njih uhvate samo jednoga i zadrže ga Ima takvih koji samo pamte gomile podataka a ima i onih koji paralelni tok podataka pretvaraju u serijski (vidi str 22—23) pa čak i takvih koji napisani tekst pretvaraju u govor (vidi str 120)

Čip sam po sebi nije posebno koristan Tek kada se napoji električnom strujom i signalima iz drugih čipova i kada su njegovi izlazi prilagođeni ostalim komponentama elektronskog lanca bit će u svojoj pravoj funkciji Spajanje čipove vrši se metalnim trakama otisnutim obično na podlozi od fiberglasa na tiskanoj pločici, kako se to stručno naziva Na stražnjoj strani pločice također su otisnuti metalni vodovi, jer sve ove komplicirane veze ne mogu stati samo na jednu stranu i sada već postaje jasno da se najveći dio posla oko stvaranja jednog kompjutora sastoji u pravilnom odabiranju čipova i pažljivom projektiranju tiskane pločice na koju će se čipovi montirati

Projektiranje novog kompjutora postaje još lakše i dakako brže ako ga obavi drugi kompjutor Posebni softver će sav taj mukotrpni posao spajanja čipove i dizajniranja nove pločice obaviti brzo sigurno i nepogrešivo

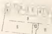
Preostaje još samo da se tehnikom nalik tiskarskoj izradi pločica da se u izbušene rupice utaknu čipovi sve skupa zalemi i ispita Cijeli taj posao može se obaviti gotovo automatski tako da je gradnje kompjutora više nalik pravljenju knjige nego na primjer izgradnji kuće



Izrade pločice poput ove na slici vrlo je jeftina i svakim danom cijena joj je sve manja Među najskuplje komponente kompjutora spada kućište s tastaturom i elektronski sklop koji kompjutor snabdjeva strujom (power supplay) Nije daleko dan kada će najveći dio cijene biti sadržan u limenom kućištu i u — troškovima transporta

Na ovoj uzorno izvedenoj tiskanoj ploči vide se svi funkcionalni dijelovi jednog mikrokompjutora. Čipovi (crni pravokutnici) su međusobno povezani metalnim vodovima (svijetle linije) otisnutim na gornjoj strani ploče u horizontalnom, a na stražnjoj strani u vertikalnom smjeru. Svijetli kružići predstavljaju metalizirane veze između njih.

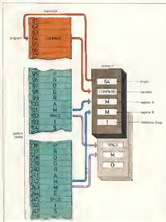
1 Generator taktne frekvencije (system clock)
2 Dekodiranje busa (bus decoding)
3 Procesor Z80
4 Direktni pristup memoriji (Direct memory access)
5 Ulaz/izlaz (input/output)
6 ROM (Read only memory — stalna memorija)
7 64 K RAM (Random access memory — radna memorija)
8 Priključak za dodatni bus (Extra bus socket)
9 Video
10 Sat korisnika (User clock)
11 Kontrola flopi diska (Floppy disk control)



MEMORIJA I PROCESOR

Za korisnika je obično najvažnija količina memorije kojom raspolaže njegov kompjutor. To se dakako odnosi na unutarnju, brzu memoriju, a ne na vanjski disk ili magnetsku vrpcu. Što više memorije — to bolje, jer će se tada moći na njemu vrtjeti veći programi i bilježiti više podataka. Zbog razloga što ćemo ih kasnije razmatrati, svi 8-bitni mikrokompjutori (u to grupu spada u ovom času najveći dio kompjutora u svijetu) imaju najviše 2^{16}, što iznosi 65536 memorijskih lokacija koje procesor može odjednom adresirati. (Zbog nekih razloga 8-bitni procesori obično upotrebljavaju 16 bita za adresiranje memorije.)

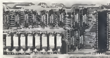
Kada bi netko sadržaj svake memorijske lokacije jednog mikrokompjutora ispisao na obične kataloške listice i poslagao ih u nizu jedan do drugog, niz bi bio dugačak oko deset kilometara. Kod 16-bitnih kompjutora udaljenost bi bila još mnogo veća, preko 160 kilometara. Zamislimo da treba trčati gore-dole uz takav beskrajni niz, pokupiti tu i tamo poneku karticu, pročitati je, pa odjuriti daleko naprijed da bi pročitali drugu koja se odnosi na prvu. Zatim odletjeti natrag, pokupiti još dvije i saopćiti rezultat. Nekome se može ovakva usporedba učiniti besmislenom, ali činjenica je da kompjutor ne radi ništa drugo do li ono što svatko može uraditi s olovkom i papirom u ruci. No kompjutor to radi neizmjerno brže i točnije — što svakako znači kvalitetan skok u radnim mogućnostima.

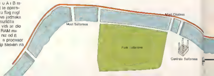


Ako procesor ima zadaću da potraži riječ PROGRAMME i zamijeni je sa PROGRAM, potražit će najprije tekuću naredbu (naredbe su locirane na početnom dijelu memorije — manji brojevi) koja glasi: USPOREDI! Kada je nađe, prenijet će je u registar naredbi, a adresu na kojoj se nalazi u memoriju programskoga broja. Slova koja treba usporediti prenijet će sada iz dijela memorije u A i B registar. Rezultat te operacije je jedinica u flag registru ako su slova jednaka, a nula ako su različita.
Na desnoj slici vidi se dio kompjutora sa RAM memorijom. To je niz od 8 svijetlih čipova, a procesor je onaj veliki čip sasvim na desnoj strani.

Srce svakog kompjutora je procesor. Kao i motor automobila, to je jedan od najvažnijih dijelova cjeline, međutim, treba još i mnogo drugih dijelova da bi cijela stvar funkcionirala. Ima raznih tipova procesora, ali svi rade na istim osnovama. Jedina i najvažnija razlika leži u duljini riječi kojom se služe. Riječ je sasvim poseban pojam u kompjutorskoj terminologiji i nema ništa zajedničkog s riječima u ovoj knjizi. Pojam riječ predstavlja osnovnu jedinicu podataka (data), kojom se procesor služi. Sada se većina mikroprocesora u svijetu služi 8-bitnim riječima. To znači da se oni hrane u 8-bitnim (ili jedno-bajtnim) zalogajima.

Sasvim pojednostavnjeno, procesor je čip s tri memorijske lokacije (samo 8 tranzistora u redu) koje nazivamo registrima. Prva lokacija sadrži naredbu, druge dvije bajtove podataka. Već prema vrsti naredbe, procesor može zbrojiti dva bajta, oduzeti jedan od drugoga ili provjeriti da li su oba jednaka. I to je sve njegovo znanje. Dakako, čak i to je vrlo, vrlo mnogo.





Promatrajući taj kratki spisak prividno nesuvislih radnji čovjek se može zapitati kakva je od njih korist. Ali treba zapamtiti da, ako je moguća operacija zbrajanja i oduzimanja (uspoređivanje je također oduzimanje s traženjem nule kao rezultata), onda je dakako moguće množenje i dijeljenje. Ako se pak može zbrajati, oduzimati, množiti i dijeliti, može se vaditi i kvadratni korijen i logaritmirati. Pa ako je sve to moguće, mogu se izvoditi i sve ostale matematičke operacije.

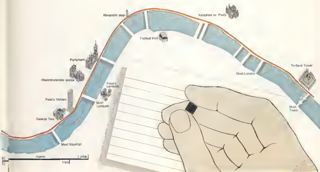
Zapravo procesor češće izvršava neke sasvim banalne radnje, kao na primjer onu preinaku riječi PROGRAMME u PROGRAM. On će izvesti uspoređujući šifre za slova MME sa šiframa ostalih slova sve dok ne nađe kombinaciju koja točno odgovara zadanoj. Tada će jednostavno izvršiti zamjenu: izbrisati MME i na tom mjestu zapisati M. Kao što smo naučili na stranici 12, sva su slova kodirana brojevima, tako da je i to samo jednostavna operacija uspoređivanja.

Pravi procesori su mnogo složeniji od ovog našega hipotetičkog procesora koji ima samo tri registra. Svi međutim rade na isti način: 16-bitni, stari 32-bitni i oni 64-bitni u velikim kompjutorskim sustavima. Isto tako radit će i novi 16, 32 i 64-bitni procesori ugrađeni u kompjutore kojima ćemo se služiti u budućnosti. Na svu sreću, mi, obični korisnici kompjutora, ne moramo razmišljati o tome kako rade procesori. To je briga onih koji se bave kompjutorskim jezicima, koji ih izmišljaju i sastavljaju. Mnogi od nas čak niti ne izrađuju sami svoje programe već jednostavno kupuju gotove programske pakete i jedini im je posao da napune svoje kompjutore i puste ih u pogon. Ljudi koji se bave sastavljanjem programa sigurno se služe nekim višim programskim jezikom, ali čak ni oni vjerojatno ne znaju baš sasvim točno kako radi procesor (vidi strojni kod i strukturirane podatke, str. 80).

Procesor nije ništa značajniji od memorije. Njih dvoje moraju raditi zajedno. Suvremeni kompjutori služe se svojom memorijom za dvije vrlo različite svrhe: za pohranu ili kako se među kompjutorašima kaže, storiranja programa, te za pohranu data.

Pretpostavimo da smo opet u potrazi za onim suvišnim M u riječi PROGRAM. ASCII šifra za slovo M je 109 (ili 01-101101 binarno) i spremljen je u A registar procesora. ASCII kôdovi ostalih slova nalaze se, recimo, u B registru. Procesor tada prima naredbu da usporedi sadržaj A i B registra. Zapovijedi što ih procesor prima dolaze iz drugog područja memorije i uvode se u registar naredbi. Ako su sadržaji A i B registra jednaki, javlja se znak. Draž ove sheme je u tome što se na taj način mogu miješati podaci i naredbe u istoj memoriji, u isto vrijeme, i u bilo kojem omjeru.

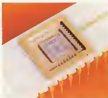
Kada bi se sadržaj svake memorijske lokacije nekog mikrokompjutora od 64 K RAM memorije ispisao na običnim kataloškim listićima kakvi se upotrebljavaju u bibliotekarstvu i kada bi se listići poslagali jedan do drugog uzduž obale Temze kod Londona, niz bi bio dugačak gotovo 10 kilometara.



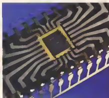


U ovom nizu fotografija prikazan je jedan integrirani sklop u plastičnom kućištu Sam čip je veličine oko 5 × 5 mm

Gore Kućište integriranog sklopa u koje je smješten čip S obje strane kućište vide se *nožice* koje služe za lemljenje čipa na tiskanu pločicu



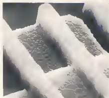
Gore sredina Procesorski čip bez kućišta



Gore desno Čip je spojen vrlo tankim vodovima na mnogo čvršće *nožice* postavljene u dva reda po jedan sa svake strane kućišta



Dolje lijevo Veliko povećanje Logički sklopovi sastoje se od nekoliko tranzistora Oni predstavljaju arhitekturu čipa i raspoređeni su u blokovima



Dolje desno Zidovi su visoki samo jednu tisućinku milimetra Cijeli čip povećan do razmjera ove slike imao bi stranicu dugačku 6 5 kilometara

Na suprotnoj strani Zemljopisna karta jednog novog svijeta procesorski čip tvrtke Texas Instruments veličine 5 × 5 mm Lijevo gore vidljivo je područje RAM memorije Gore desno aritmetičko-logička jedinica Dolje lijevo područje ROM memorije za dekodiranje naredbi a dolje desno procesor koji se sastoji od dvije grupe po 8 registara Oko vanjskog ruba vidljivi su kontakti (svijetle četvorine) na koje se pričvršćuju vodovi nožica Pokraj svake je vidljiv još po jedan elektronski sklop za dekodiranja

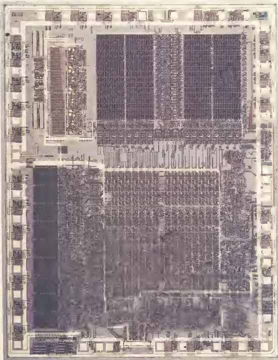
Kao što će se vidjeti na stranicama 165-6 s nekoliko tranzistora može se izvesti svaki logički zaključak Pa ako se to može moguć je i svaki drugi logički specificirani postupak koliko god bio zakučast i složen Na primjer nema zaista razloga da čovjek ne sagradi od tranzistora naprave koje bi mogle vaditi iz utrobe zemlje materijale iz kojih bi oni sami sebe reproducirali Bila bi to neka vrst elektronskog virusa koja bi se mogle posIati na put kroz svemir pa gdje god bi naišla na pogodno tlo stvarala bi vlastite kopije Kad bi te naprave bile snabdjevene još i nekom vrsti rudimentarnog nagona samoodržanja i zapažanja eto naprave koja bi mogla kolonizirati svemir Kasnije dakako, mogli bismo požaliti vlastitu preuzetnost i neoprez

No zaboravimo tehničku fantastiku i usmjerimo svoju pažnju na tranzistor Kao što se vidi na slikama tranzistor je načinjen od nekih jastučastih nabora To su vodovi kojima se prenosi elektricitet Na sastavu vodova nastaje tranzistor koji pojednostavnjeno gledano možemo smatrati jednostavnim elektronskim preklopnikom

Da bismo stekli pojam o veličinama elektronskih elemenata kojima su napunjeni čipovi treba shvatiti da su jastučaste izbočine (na posljednjoj fotografiji) široke oko dva milijunta dijela metra (2 mikrona) To znači kad bi se primijenile iste tehnologija (izrade standardnog 8-bitnog mikroprocesora) pri izradi zemljopisnih karata mogao bi se na veličini od 5 × 5 mm prikazati cijeli plan londonskog središta sa svim ulicama i trgovima ili sabiti 15 000 riječi nekog teksta Na površinu od četiri ovakva čipa stao bi cijeli sadržaj ove knjige Još savršenijom tehnologijom kakvom se izrađuju novi 16- i 32-bitni procesori, moglo bi se na njihovoj površini prikazati plan cijelog Londona New Yorka Pariza ili Moskve Začuđuje spoznaja da možemo izraditi stvari tako velikih mogućnosti a ipak tako sitne da se mogu zagubiti u džepiću prsluka

Tehnologije za izradu kompjutora u idućih pet godina još su u fazi istraživanja a bit će tako savršene da će na površini od 5 × 5 mm stati toliko detalja koliko ih ima na zemljopisnoj karti Kalifornije od San Franciska do Los Angelesa

Danas je već sasvim jasno da su te sićušne stvari najsloženije naprave koje je čovjek ikada stvorio a proizvode se u tako velikim serijama da im cijena ne prelazi vrijednost od nekoliko kutija cigareta



TRANZISTORSKA LOGIKA

Faze u proizvodnji tranzistora

A

B

C

D

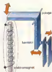
E

sors

drejn

F

Svaki tranzistor služi upravo onome čemu služi i stari dobri relej poput ovoga što je prikazan na slici desno. Sastoji se od željezne jezgre oko koje je namotana žica. Kada se na izvode namotaja priključi električna struja, ona će poteći kroz njega, stvoriti magnetsko polje i privući kotvu. Kontakti na njenom kraju spojiti će se, a posljedica je toga da se slabom strujom (što teče kroz magnetov namotaj) može ukopčati neko veće trošilo čija će struja teći kroz kontakte releja.



Isto radi tranzistor, ali je neusporedivo manji i mnogo jeftiniji.*

Osnovna sirovina za izradu tranzistora je silicij što ga ima u svakom kamenu i pijesku (čini oko 25 % zemljine površine). Topljenjem i pročišćavanjem prave se okrugle pogače iz kojih se izrezuju pločice čistog silicija.

Silicij se može oksidacijom pretvoriti u staklo i od njega napraviti električki izolator, mogu se u njega utaliti aluminijski vodovi koji će poput žica provoditi elektricitet, a može ga se bombardirati atomima joda (ili ga nekako drugačije onečistiti), pa ga natjerati da u nekim okolnostima provodi struju, a u nekim opet ne. Ta mu je osobina darovala ime: poluvodič.

Na slici lijevo prikazana je proizvodnja tranzistora koja se odvija u četiri faze. Najprije se pločica silicija presvuče slojem oksida (A), zatim se popriječko izdubi žlijeb i na tom mjestu skine oksid tako da na sredini ostane samo mali mostić (B). Ostatak okside se ispere (C). Nakon toga se nanese tanak sloj aluminija na mostić i sve zajedno odmbardira atomima joda — taj dio postupka naziva se dopiranje — (D).

Takav tranzistor ima tri izvoda: gejt, sors i drejn. Nazivi su tradicionalni, iako bez nekog posebnog smisla**, ali je zato smisao tranzistora i te kako važan.

Elektricitet bi mogao nesmetano teći kroz dopirani silicij da mu na putu nije postavljena brana u obliku aluminijskoga mostića. Zešticen aluminijem, silicij nije mogao biti dopiran, pa se ponaša kao izolator sve dok se oko njega ne pojavi električko polje. Ako se preko gejta dovede na mostić nekakav napon, struja će poteći od sorsa prema drejnu (E). Ako se pak napon isključi, proces se prekida (F). Na taj način stvorena je vrlo brza, električki kontrolirana sklopka i još jedno iznenađenje što nam svakodnevno priređuju poluvodiči.

Logički sklopovi

Tranzistorska sklopka što smo je maloprije opisali nije naročito korisna, kao što ni obična opeka sama za sebe ne znači ništa. Ali od mnogo malih opeka može se sagraditi veliki neboder. Isto je tako i sa tranzistorima. Organizirani u grupe mogu izvoditi vrlo važne logičke operacije.

Na crtežu F prikazan je jednostavni logički sklop tzv. invertor ili NE (NOT) sklop. Tako je nazvan jer mu izlaz nikada nije jednak ulazu. (Naime, ako je na sors preko jednog otpornika priključen napon od 5 volti — to je standardni napon s kojim rade logički sklopovi — a drejn uzemljen, i kada se isti napon priključi na gejt, struja će isteći brže kroz drejn nego što može pritjecati preko otpornika do sorsa, pa će tada napon na sorsu imati vrijednost 0.) Ako se logički 1 nalazi na ulazu, na izlazu će uvijek biti logička 0 i obratno: 0 na ulazu, 1 na izlazu — stalna negacija. To je po prilici sve što se može učiniti s jednim ulazom. Ako se pak naprave dva ulaza i jedan izlaz, a svaki od njih može biti 1 ili 0, mogu se stvoriti tri logička sklopa: I (AND), ILI (OR) i ISKLJUČIVI ILI (XOR).

Ove tri operacije prikazane su na tzv. »tabli istinitosti«. U dvije kolone s lijeve strane prikazane su moguće kombinacije dvaju ulaza A i B. Na kolonama s desne strane prikazani su odgovarajući izlazi. (Upotreba riječi istinitost ostatak je iz onih vremena kada su se time bavili samo logičari.)

NE (NOT)

ulaz	izlaz
1	0
0	1

I (AND)

ulaz A	ulaz B	izlaz
1	1	1
1	0	0
0	1	0
0	0	0

ILI (OR)

ulaz A	ulaz B	izlaz
1	1	1
1	0	1
0	1	1
0	0	0

(XOR)

ulaz A	ulaz B	izlaz
1	1	0
1	0	1
0	1	1
0	0	0

U AND logičkom sklopu, na primjer, izlaz je 1 ako su oba ulaza 1. U svim ostalim slučajevima je 0. U OR logici izlaz je 0 ako su oba ulaza 0, u

* Sva daljnja razmatranja o tranzistoru ne odnose se na klasične tranzistore, već na tranzistore izvedene u MOS FET (Metal Oxide Semiconductor — Field Effect Transistor) tehnologiji, jer se upravo ova upotrebljava u proizvodnji tranzistora kao osnovnih sastavnih dijelova logičkih sklopova u impulsnoj i digitalnoj elektronici. Prim. prev.

** Za razliku od klasičnog tranzistora, čije izvode nazivamo baza, emiter i kolektor, izvodi na fet tranzistoru, a takvi se isključivo primjenjuju u tzv. LSI tehnologiji (Large Scale Integration — visoki stupanj integracije), nazivaju se gate (vrata), sorce (izvor) i drain (kanal). Te nazive ne bi imalo smisla prevoditi jer su među hardverašima ionako udomaćeni engleski izrazi, pa ćemo ih, kao uostalom i mnoge druge u ovoj knjizi, i mi prihvatiti pišući ih dakako fonetski. Prim. prev.

svim ostalim slučajevima je 1 U XOR logici izlaz je 0 ako su oba ulaza jednaka a ako su različita onda je 1

Budući da samo jedan tranzistor invertira signal najlakša izvedba je NAND i NOR logike (na gornjim tablicama prikazani kao AND i OR logika s invertiranim izlazima — na izlazu se zamijenjuje i sa 0 i obratno)

Ako su oba ulaza NAND logike 1 izlaz može biti samo 0 Izlaz NOR logike je uvijek 0 ako je jedan ili oba ulaza 1 U svakom drugom slučaju je 0

Takva dva logička sklopa mogu se izvesti (kako se vidi na slici desno) uz pomoć vodovodnih cijevi i vode Voda će teći samo kroz AND cijev ako su obje slavine otvorene Iz OR cijevi naprotiv voda će teći ako je samo slavina A ili samo slavina B otvorena

Ova razmatranja sama za sebe mogu nam se činiti beskorisnima Ali ako se sjetimo onih triju operacija (str 16-17) koje 8-bitni procesor može izvesti vidjet ćemo da i nisu tako beskorisna Značajno je što postoje dva registra a jedna od mogućnosti je u tome da procesor može usporediti da li oba registra sadrže isti 8-bitni bajt To izgleda ovako

SADRŽAJ REGISTRA A	1 0 1 0 1 1 1 0
SADRŽAJ REGISTRA B	1 0 1 0 1 1 1 0
XOR A B	
SADRŽAJ REGISTRA C	0 0 0 0 0 0 0 0

Zaključak oba bajta su jednaka

To se može izvesti i na drugi način tj da se načini kombinacija koja obrađuje prvo po XOR logici po dva bita u paru a zatim po OR logici također bitove u parovima u kaskadi Rezultat će biti 0 ako su bitovi jednaki a 1 ako su različiti

```
A      1 1 0 1 0 1 1 0
B      1 1 0 1 0 1 1 0
XOR    0 0 0 0 0 0 0 0
OR      0   0   0   0
OR        0       0
OR            0
```

Zaključak A i B su jednaki

Pogledajmo kako izgleda kada su A i B različiti

```
A      1 1 0 1 0 1 1 0
B      1 1 1 0 0 1 1 0
XILI   0 0 1 1 0 0 0 0
ILI     0   1   0   0
ILI       1       0
ILI           1
```

Zaključak A i B su različiti

I evo sada se već nazire svrha i logike logičkih sklopova

Moguće je također napraviti integrirani sklop koji će vršiti 8-bitnu komparaciju funkcija procesora. Pokušajmo stoga zbrajati bitove, jer se često događa da pri rješavanju logičkih problema pomaža ispisivanje svih mogućnosti

A	B	odgovor	prenos
0 +	0 =	0	0
1 +	0 =	1	0
0 +	1 =	1	0
1 +	1 =	0	1

Odgovor je dobiven XILI logikom ulaza A i B a prenos po I logici Ovo se može proširiti zbrajanjem dvaju 8-bitnih brojeva

PRENOS	0 1 1 1 1 0 1 0
REGISTAR A	0 0 1 1 1 0 1 0
REGISTAR B	0 1 0 1 1 0 1 0
REGISTAR C	1 0 0 1 0 1 0 0

Ovo i nije tako jednostavno ali je još uvijek podnošljivo Idući odozgo prema dolje ima tri bita koje treba zbrojiti Bit A bit B i prenos od prethodnog zbrajanja (koji ima vrijednost 0 u sasvim desnoj koloni) Bitovi se zbrajaju upravo onako kako smo učili u školi A + B + C je isto kao kad izrazu A + B dodamo C

Ako ste još uvijek svježi predlažem da sami načinite jednu tablicu s odbijanjem Nakon toga mora nam biti jasno da možemo uspoređivati zbrajati i oduzimati a također množiti i dijeliti Ako pak možemo množiti i dijeliti možemo rješavati jednadžbe baviti se statistikom i predviđati rezultate nekih složenih zbivanja I da se vratimo logičkim funkcijama ako njih možemo rješavati tada se u različitim kombinacijama može izvesti svaka operacija koja se može definirati kao niz logičkih koraka Kao što će se vidjeti to nije lako ali čovjek je pametniji nego što pretpostavlja

I (AND) — otvoreno, otvoreno, A, B

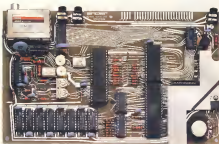
ILI (OR) — otvoreno, A, zatvoreno, B

Princip rada logičkog I ILI (AND i OR) sklopa može se lako shvatiti na primjeru vodovodne cijevi i slavina Ako su (kao na gornjoj slici) slavine A i slavine B otvorene — voda teče Ako su obje (ili samo jedna) zatvorene — vode nema Na donjoj slici ako je slavina A ili B otvorena voda teče

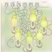
NITI (NAND) — NILI (NOR)

Na shemi s lijeve strane ako su tranzistor A i tranzistor B provodni struja teče kroz njih i napon na izlazu pada na nulu Na desnoj shemi ako je tranzistor A ili tranzistor B provodan struja će teći i na izlazu je opet vrijednost 0 Kako su u obe spoje izlazi inverzni ulazima ovi se logički sklopovi nazivaju NITI i NILI (NAND i NOR)

BUSOVI ILI SABIRNICE



Tiskana pločica Sinclairove Spectruma. Dolje lijevo RAM čipovi, dolje desno Z80 procesor, gore desno ROM (u koji je smješten BASIC i upravljački sustav). Kontakti za napajanje radnim naponima vide se uz gornji rub ploče, a u sredini je ULA (Uncommitted Logic Array — polje neizvršenih naredbi) koje nadomješta niz pojedinačnih integriranih sklopova. Sve je to povezano u jednu cjelinu BUSOM čije linije uglavnom teku stražnjom stranom ploče.





Ako svaku svjetiljku posebno povežemo s izvorom napajanja (gore), gazit ćemo preko nerazmrsivog mnoštva žica. Bolje je voditi samo jednu žicu (dolje). Na sličan način funkcionira bus.

Da bi kompjutor mogao funkcionirati, potrebno je mnogo sklopova povezati u suvislu i čvrstu cjelinu bez obzira na to da li su to čipovi na tiskanoj pločici, nekoliko tiskanih ploča u jednom kompjutoru, kompjutori u lokalnoj kompjutorskoj mreži ili kompjutorske mreže u nacionalnom ili internacionalnom sustavu. A mogu to biti i radari ili oružja na ratnom brodu, komande i veze u avionu ili robotu, ili strojevi i senzori u nekoj automatiziranoj tvornici. Netko će pomisliti da je najjednostavnije komadom žice povezati glavno upravljačko mjesto sa sporednim i obratno, sporedno s glavnim. Takvo rješenje je moguće, ali ima mnogo mana. U prvom redu za to treba mnogo žica i sva sila priključaka. Dodavanjem svakog novog člana namnožilo bi se toliko priključaka i žica da se više nitko u tome ne bi mogao snaći.

Rješenje je tzv. BUS, jedna jedina linija koja obilazi sve članove cjeline. U modernoj zgradi električke linije u prstenovima obilaze katove (svaki prsten je jednostavan primjer busa) i dovode struju do utičnica u svakom stanu. Tako su i čipovi u kompjutoru povezani busovima preko kojih primaju energiju, naredbe i raznose podatke.

Razne periferije ne djeluju istovremeno i ne rade iste stvari. Ako su svi čipovi i svi priključci povezani istim komadom žice, mora postojati neki uređaj koji će nadgledati cijeli sustav i naređivati pojedinim sekcijama i periferijama kada da prorade. Premda bus istodobno sve povezuje, njegova suština je u tome da svaka naprava prima ili šalje informacije preko njega i to onda kada joj je naređeno.

Busovi koji vode izvan kompjutora nazivaju se PORTE (vidi str. 28—29). Ima ih dvije vrste: serijske i paralelne. Usporedimo li ih s mostom, najlakše ćemo shvatiti razliku. Zamislimo kolonu po osam vojnika u redu (paralelno) koja marširа prema mostu. Konstruktor mosta imao je dvije mogućnosti: mogao je sagraditi skupi široki most, tako da se vojnici ne moraju prestrojavati, već ga mogu preći po osmorica u redu (paralelno). A mogao je sagraditi i uski jeftini most koji bi prisilio vojnike da se prestrojavaju i prelaze jedan po jedan (serijski). Konstruktor busa stoji pred istim problemom: zadatak mu je da za određeno vrijeme prenese izvjestan broj bitova s jednog mjesta na drugo. On ih može slati jednog po jednog, serijski, samo kroz jednu žicu, ili paralelno, po nekoliko njih istodobno, ali kroz više žica.



Podaci putuju unutar kompjutora paralelno. Serijski bus izlazi iz posebnog čipa (SIO — serial input/output — serijski ulaz/izlaz) koji odjednom prima po 8 bitova paralelno, a očitava ih jednog po jednog. To zahtijeva više čipova u kompjutoru i dakako skuplje je, ali traži manje žica. To je razlog zbog kojeg su skuplje one vrste periferija (npr. printeri) koje serijski primaju podatke od onih koje ih primaju paralelno.

Ali kada zaista treba prenositi podatke velikom brzinom, na primjer između diska i proce-

sora, konstruktori će takav spoj uvijek izvesti paralelnim busom

Pogledajmo sada jedan elektronski ured u ne tako dalekoj budućnosti. Helen, djevojka u prednjem planu, treba da pošalje neki podatak sa tvrdog diska (hard disk) da bi se nešto u njemu promijenilo. Ona otipkava poruku kojom najavljuje svoju namjeru. Njezin kompjutor čeka sve dok glavna stanica ne završi posao koji upravo radi i tada počinje radnim mjestima dojavljivati seriju poruka: broj 1, treba li štogod? broj 2, treba li štogod? Čim se začuje poziv svi ostali zašute da ne bi smetali glavnoj stanici. Naša Helen je na broju 3 i kada na nju dođe red stanica prima njezin zahtjev. Sad glavna stanica prekida brbljanje da bi poručila hard disku neka čita sadržaj što ga broj 3 očekuje. Dok hard disk čita, kompjutor broj 3 pažljivo osluškuje i kako se tekst pojavljuje, odmah ga kopira u svoju memoriju da bi ga Helen mogla obrađivati. I tako neprestano traje taj nečujni proces koji se ovako prepričan čini zamršenim, ali se zapravo odvija tako brzo da ljudi za tipkama kompjutora nisu toga ni svjesni.

Moderni ratni brod je dobar primjer za primjenu busa. Samo jedna linija optičkih vlakana (dvostruka ili trostruka kao osiguranje od oštećenja u bitki) povezuje sva vitalna mjesta broda. Takav bus zamjenjuje kilometre teških i ranjivih kablova, neosjetljiv je na električke interferencije i jednostavnije se održava.

Službenici u uredu bliske budućnosti služe se za izmjenu informacija kompjutorima međusobno povezanim busom (točkasta linija).

MEMORIJA

RAM memorija u kompjutoru podijeljena je na oštro omeđena područje od kojih svako obavlje specifične zadatke U ovom primjeru prvo područje je zauzeto za grananje programa (operating system jump tables) i još za neke sitnice Slijedeće područje sadrži softver za izlaz na monitor Nakon toga dolaze tri područja TPA (Transient Program Area — prolazni programski prostor) gdje su stacionirani privremeni programi Prvi dio tog prostora zauzima programski jezik BASIC u drugom su programi koji se na tom jeziku odvijaju a u trećem je STACK BASICa Ovaj prostor može zauzimati i svaki drugi programski jezik Na kraju je područje s kôdom za upravljanje diskom monitorom i izlazom za printer

Dovede li se pozitivni napon na gejt tranzistora sors-drejn pravac postaje vodljiv i struja nesmetano otiče s donjeg kraja otpornike Na sorsu napon tada iznosi 0 volti i on je kako se kaže L (low) Ako je pak na gejtu 0 volti struja ne teče kroz tranzistor i sors je sada H (high)

Statička memorijska ćelija sastavljena od dva tranzistore Jedan od njih je uvijek provođen a drugi zakočen U njemu je dakle uvijek spremljena 0 ili 1 (Na slici je izlaz A plav — L ili O) Oba tranzistora su tako povezane da jedan drugog drže u određenom stanju

Ako procesor predstavlja motor kompjutora, memorija je karoserija u kojoj se programi voze Ta živa vrst memorije naziva se RAM (Random Access Memory) memorija u koju možemo upisati ili iz koje možemo direktno bez ikakvih priprema pročitati bilo koji bajt Memorija sa stalnim sadržajem naziva se ROM (Read Only Memory — memorija koje se samo čita) Između tih dviju memorija postoje još i različite vrste PROM memorija (Programmable Read Only Memory — programabilni ROM)

Kao što smo se već uvjerili što više memorija u kompjutoru to bolje a kao što ćemo se još uvjeriti količina memorije ovisna je o cijeni kompjutora a ona opet o gustoći memorijskih čipova (količini elektronskih elemenata koja se može sabiti na komadić silicija) koju proizvođač može postići svojom tehnologijom

Pokušajmo sada objasniti kako se može komadić silicija natjerati da bilo što upamti Na strani 20 utvrdili smo da je tranzistor u prvom redu elektronski preklopnik pa ako mu dovedemo napon na gejt struja će kroz njega poteći Prekinemo li je tranzistor će se kako se to stručno kaže zakočiti U tome dakako nema ništa uzbudljiva ali ako povežemo dva tranzistora tako da je njihova vodljivost međusobno ovisna napravili smo ono što stručnjaci nazivaju flip—flop

Na slici dolje lijevo shematski su prikazana dva tranzistora sa sorsovima priključenim na pozitivni napon preko dva otpornika Drejnovi su uzemljeni a svaki gejt jednog tranzistora povezan je sa sorsom drugog Pretpostavimo da je prvi tranzistor provodan Struja teče kroz tranzistor brže nego što joj dozvoljava otpornik tako da na njegovu sorsu vlada napon 0 Budući da je gejt drugog tranzistora vezan na sors prvoga drugi je zbog toga neprovodan struja teče kroz njegov otpornik ali kako ne može dalje otjecati na donjem njegovu kraju (točka B) je napon (H) visok Gejt prvog tranzistora priključen je na taj visoki napon provodan je i na točki A vlada napon 0 (L) I — našli smo se ondje odakle smo krenuli!

Karakteristično je za takav spoj da on ostaje u početnom stanju dok se situacija na ulazu ne promijeni To znači da će točka A stalno imati vrijednost 0 (ili L low — nizak) a točka B će stalno zadržavati najvišu vrijednost ulaznog napona predstavljenu sa 1 (ili H, high — visok) Kada stanja na ulazu sklopa promijenimo i tranzistori će tada uzajamno zamijeniti uloga

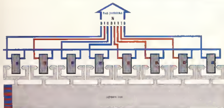
Niski napon ili 0 na ulazu sklopa učinit će da na točki A bude visoki (H) napon ili 1 I obratno visoki napon (H) na ulazu natjerat će točku A da promijeni stanje i da bude niske (L) vrijednosti ili 0 To je flip ili flop jedan ili nula H ili L Posljedica toga je pamćenje memorija

Sve to ne bi imalo mnogo smisla kad se ne bi moglo pisati u ovakvu memoriju a to se čini upravo na maloprije opisani način I upravo tako izgleda jedna memorijska celija s tom razlikom što se na mjestu otpornika nalazi još jedan tranzistor čiji je drejn povezan s vlastitim gejtom (slika dolje) te tako obavlja funkciju otpornika što je u tehnici integriranih sklopova teže izvesti nego tranzistor

Svaki flip—flop pamti po jedan bit informacije Da bi se postigla upotrebljiva količina memorije nekoliko tisuća takvih flip—flopova utiskuju se jedan do drugog na komadić silicija Ta vrst memorija naziva se statičkom za razliku od dinamičke' o kojoj ćemo raspravljati u slijedećem odjeljku

Samo po sebi je jasno da četiri tranzistora (koliko ih ima u sklopu što smo ga već opisali) zauzimaju četiri puta više mjesta od jednog tranzistora To znači da bi na isti prostor stalo četiri puta više podataka kada bi se memorija sastojala od samo jednog tranzistora To nije izvedivo, ali je ipak moguća







Adresni bus memorije (na slici lijevo) ima 16 linija. Uzorak binarnog broja na njemu (prikazan crvenim i plavim prugama na krajnjem lijevom kraju busa) saopćava nizu od 8 čipova, od kojih svaki sadrži 64 K bita, koju će celiju otvoriti. Izlazi svakog čipa (gore desno) povezani su s jednom od 8 linija podataka, tako da se kompletan bajt (8 bita) pojavljuje u ispravnom nizu na busu podataka. Ovdje je to broj 01001110.

kom izbrisati i upisati neki novi program ili podatak

memorija koja se sastoji samo od dva tranzistora, a može se izvesti tako da se tranzistor kontrolira preko gejta. Teorijski, to je zatvoreni krug i ne vodi nikamo, jer električni naboj doveden na tranzistor nema kamo otjecati. U principu, to je opet memorija. Možemo lako ustanoviti što je upisano u nju: 0 ili 1, ovisno o stanju koje vlada na liniji sors—drejn.

U stvarnosti elektroni se zadržavaju na gejtu vrlo kratko, ne duže od 1/1000 sekunde, no to je ipak dovoljno da centralni procesor obavi nekoliko tisuća operacija. Projurit će kroz sve svoje memorijske ćelije, pročitati sve što tamo nađe i zapisati iznova što je pročitao, tako da to ostane zapisano slijedeću tisućinku sekunde. To nazivamo „osvježenjem memorije".

Svaki memorijski čip ima izvjesten broj nožica (pinove) kojima je vezan za ostale čipove. Broj nožica ovisan je o količini memorije koju sadrži. Mora imati dovoljno adresnih linija (adress line) da može dospjeti do svake memorijske ćelije, jednu 'čitaj/piši' (read/write) liniju, koja će obavijestiti treba li nešto pročitati ili zapisati, i jednu liniju za podatke (data line), koja je povezana sa izlazom svake pojedine ćelije.

Ako neki čip ima 2^n memorijskih celija, trebat će mu n adresnih linija. Na slici gore desno prikazan je moderni čip od 64 K RAMe*. 64 K iznosi 65536 ili 2^{16}. Stoga takav čip mora imati 16 adresnih nožica.

Treba još naglasiti da jedan bajt, što u većini slučajeva predstavlja najmanju memorijsku jedinicu, sadrži 8 bitova, odnosno 8 opisanih elektronskih sklopova. Zato je spretnije da se svaki od 8 bitova jednoga bajta nalazi u posebnom čipu i da svaka linija podataka (data line) bude povezana sa po jednom linijom busa podataka (data bus).

Ima još nekoliko vrsta memorije. ROM memorija je mnogo jednostavnije. U nju se podatak upisuje samo jedanput kada se čip proizvede i ne može se više izbrisati, što se postiže ugradnjom posebne maske. U takav čip može se spremiti gomila podataka ili programa, a procesor ga shvaća kao i svaku drugu RAM memoriju. Mnogi kompjutori imaju posebne ROMove od po nekoliko kilobita koji kontroliraju njihove tastature. ROM je sastavljen od vrlo mnogo jednostavnih elektronskih sklopova koji mogu stati na mali prostor. (Japanci su nedavno napravili ROM od 256 K i u 8 takvih čipova može se spremiti oko 40 000 riječi, ili dvije trećine ove knjige). Nezgodno je što treba naručiti nekoliko tisuća komada ROMova, da bi se isplatila izrada skupe maske, koja je važan dio njegove konstrukcije. Alternativa koja je u prodaji skuplja, ali u proizvodnji ekonomičnija, je PROM (Programmable Read Only Memory). To je prazna memorija u koju se upisuju podaci ili programi tako da se propali ili ne propali jedna veza sa svakom memorijskom lokacijom, što se izvodi pod kontrolom kompjutora uz pomoć visokog napona.

Još bolje rješenje, ali dakako i skuplje, je memorijski čip EPROM (Erasable Programmable Read Only Memory), vrsta RAMe koja jednom programiran ne treba više osvježavanja. Sve što je upisano može se posebnim postup-



EPROM (Erasable Programmable Read Only Memory — programabilna samo čitaj memorija koja se može brisati) se sastoji od čipova koji se mogu naći u gotovo svakom kompjutoru. U njih se upisuje 'firmver' (firmware). Firmver je zapravo softver koji se upisuje u hardver i služi za kontroliranje osnovnih zadataka. Služeći se EPROM čipovima, proizvođač i korisnik može izbrisati svaki upisani program i preraditi ga po slobodnoj volji. Upisani podaci brišu se pomoću snažne ultraljubičaste svjetlosti kroz prozirna prozorčiće kojima su prekriveni čipovi.

* K ili prefiks kilo predstavlja kompjutorsku tisuću koja ne vrijedi 1000, već 1024, što je isto kao 2^{10}.

TASTATURA

Prosječni vlasnik ostveruje fizički kontakt sa svojim kompjutorom uglavnom preko tastature. Pedesetak tipaka poput onih na pisaćem stroju služe za najrazličitije svrhe. Pomoću njih može se pisati svaki tekst, mogu se utipkavati brojevi, tipkama se upravlja kursorima (cursor) po ekranu, preko njih se bori sa svemirskim osvajačima, piše poezija, crta ili vodi računovodstvo.

Ne postoji neki čvrsti standard za kompjutorske tastature, ali ipak u većini slučajeva na njima će se naći sljedeće funkcije:

Slova abecede sa šift tipkom (shift—pomicanje) kojom se tastatura prebacuje s malih slova na velike i obratno.

Pračka za otipkavanje razmaka (space bar).

Uobičajeni simboli * $ % &

Kolekcija zagrade [] () {}

Aritmetički simboli + — * / ↑ < > to jest znakovi za zbrajanje, oduzimanje, množenje, dijeljenje, potenciranje i < (manje) odnosno > (veće).

Interpunkcijske oznake ? !

Dvije vrste navodnika ' "

Simboli uobičajeni na kompjutorima a rijetko se nalaze na pisaćim strojevima ~ | \

Četiri tipke za upravljanje kursorom: gore, dolje, lijevo ili desno.

Uza sve to naći ćemo još barem četiri posebne tipke DELETE, CONTROL, ESCAPE i RETURN. One su vrlo važne pa ćemo ih potanje objasniti.

DELETE (brisati) služi onome što bi svatko želio imati na pisaćem stroju: mogućnost da obriše sve što je pogrešno napisano.

CONTROL ili CTRL (kontrolirati). Pritiskom na ovu tipku neće se pojaviti nikakav znak na ekranu. Ona se uvijek kombinira s još nekom tipkom mijenjajući značenje ove druge (vidi ASCII kodove str. 185).

ESCAPE (izlaz). Služi na nekim tastaturama da bi se iz jednog operativnog moda (načina) prešlo u drugi.

RETURN (povratak). Na nekim tastaturama označen kao NEWLINE (nova linija) ili ENTER (ući, ulaz). Tipka koja se najčešće upotrebljava i u osnovi ima istu svrhu kao i poluga na pisaćem stroju kojom se valjak vraća na početak novog reda. Uz ovu funkciju return ima još i značenje naredbe obavi ili izvrši. U programu nakon što smo otipkali neki broj ili naredbu, procesor čeka signal ove tipke koji mu javlja da je sve što se od njega tražilo otipkano i da može preći u izvršavanje.

Na skupljim tastaturama postoji još i posebni numerički blok s desne strane normalne tastature koji služi za utipkavanje više brojeva. Često se kod dobrih strojeva u gornjem redu tipaka nalazi i niz neoznačenih takozvanih "soft-key" tipaka (mekana tipka). Te se tipke mogu slobodno programirati i dati im onu svrhu za koju bi inače trebalo otipkati nekoliko znakova. Na primjer, u programu kojim se služim pri pisanju ove knjige (word-processor), kada želim kompjutoru narediti da potraži neku riječ u tekstu i zamijeni je s drugom, moram redom pritisnuti tri tipke ESR ↑ i R. Kad bih imao slo-



bodno programabilne tipke jednostavno bih kombinirao sva tri znaka u jedan i sve spremio u RAM tastature. Tada bi bilo dovoljno pritisnuti samo jednu tipku za ovu složenu naredbu. Pretaskom na neki drugi program mogu svakoj od ovih tipaka dati novo značenje.

Kada kompjutor ne može primati naradbe tastature jer je zauzet nekim drugim poslom, dobra tastatura otipkane poruke sprema na neko vrijeme u za to rezervirani dio RAMa i šalje ih procesoru kad bude slobodan (takav uređaj zove se type-ahead ili rollover'). Mnoge tastature oponašaju električne pisaće strojeve pa ako tipku držite pritisnutu dulje od pola sekunde njen znak će se ponavljati na ekranu sve dok je tipka pritisnuta.

Poboljšanje koje nije potrebno (barem s mojeg stajališta) je onaj klik koji se čuje na pisaćem stroju keda se otipka slovo. Neki proizvođači do te mjare pokušavaju imitirati pisaće strojeve da ugrađuju u tastaturu mali zvučnik koji se javlja uz pritisak na svaku tipku. Bolje bi bilo da zvučnik javlja nekim signalom počinjenu pogrešku.

Vidjeli smo već na stranama 12-13 da su svi znakovi tastature kodirani ASCII šiframa (tablica na strani 185.). U tom sustavu ima nekoliko vrlo oštroumnih rješenja. ASCII šifra za veliko A (65), na primjer, je za 32 manja nego za malo a (97). Isto tako je B (66) za 32 manja od b (98). Treba dakle uvijek šifri dodati 32 da bi se velika slova konvertirala u mala.

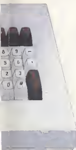
Jasno je da brojevi rastu usporedno s abecedom. Isto je i sa šiframa brojeva. ASCII kód za 1 iznosi 49, a za 2 50. Da bi se napravila tablica



Pritiskom na neku tipku tastature nastaje spoj između vodova koji teku ispod horizontalnih redova tipaka i onih ispod vertikalnih kolona. Tako dobiveni signali dovode se do ROMa koji ih prevodi u ASCII kodova. Na slici je dan signal za slovo N čija je šifra 01001110 ili decimalno 78.

ASCII kódova treba ići redom i svakom većem broju ili sljedećem slovu abecede dodati 1.

ASCII kodovi rastu do broja 127. Mnogi početnički kompjutori nude različite satove grafičkih znakova, kvadratiča, kružica i slično, koji se otipkavaju pomoći SHIFT tipke. Bolji kompjutori troše taj prostor za posebne znakove kojih ima u nekim evropskim jezicima, kao što je preglašeni a (a) ili u (u).



Sinclairov ZX Spectrum. Oštroumna konstrukcija omogućava višestruku upotrebu svake tipke. Tojest svakom se tipkom može generirati neko slovo ili broj, ali i mnoga naredbe BASICa. Tipke za grafiku i boju označene su iznad tipaka s gornje strane.
Spectrum i njegovi prethodnici ZX80 i ZX81 vjerojatno su najrasprostranjeniji kompjutori u svijetu: prodani su u više od milijun primjeraka, a cijene im iznosi samo dio nekog drugog kompjutora koji se manje prodaje. Kupuje se masovno, ili zbog znatiželjnosti, ili kao osnovni dio osobne kompjutorske opreme. Iz takvoga korisničkog sloja izrasta i novi sloj kompjuteriziranе populacije.



PORTE

Kompjutor je sâm za sebe prilično jadna i beskorisna stvar. Na njemu ćete naći preklopnik koji ga može uključiti ili isključiti i, ako imate sreće, svjetalce koje pokazuje da li radi.

Da postane upotrebljiv moral ćete mu priključiti barem još tastaturu, ekran i printer. Često ćete osjetiti da je potrebno priključiti mu još neke stvari°, npr. palicu (joystick) da njome vodite svemirce po ekranu, kuglu (roller ball) ili miša za udobnije upravljanje kursorima, ili grafički tablet da preko njega unosite crteže u memoriju kompjutora. Možda ćete zaželjeti još egzotičnije izlazne jedinice nego što je običan printer, pa će vam trebati da priključite ploter, možda laser ili robot koji može umjesto vas obavljati neke dosadne kućne poslove.

Kad bi svaka od tih naprava tražila neki svoj vlastiti priključak, rad na kompjutoru postao bi nemoguć. Da bi se to pojednostavnilo, izmišljeno je još nekoliko standarda za ono što u praksi nazivamo "portama". (Taj je naziv vjerojatno došao otuda što kompjutor preko njih komunicira s "vanjskim svijetom", s periferijama koje se nalaze izvan njega.) U osnovi postoje dvije vrste porti: serijske i paralelne. Kao što je opisano na stranama 22-23, serijska porta šalje ili prima bitove jednoga bajta, jednoga po jednog, samo preko dviju žica. Paralelna porta, naprotiv, odjednom šalje ili prima 8 ili više bitova i za to joj treba onoliko žica koliko ima bitova s kojima istovremeno radi.

Od porte na kućištu kompjutora vidi se samo podnožje utikača, najčešće žensko s 25 iglica (pinova). Sada treba još saznati da li se radi o paralelnoj ili serijskoj porti, da li odgovara serijskom standardu koji nosi oznaku RS 232, ili o paralelnom s oznakom Centronics ili IEE 488 (postoje dakako još neki standardi, ali za početnika je dovoljno i ovoliko). Ako je porta serijska, treba još znati s koliko bitova u sekundi prijenose treba računati ("Baud rate") i da li se rukuje (handshaking) ili očekuje neki drugi protokol.

Ako je periferija na većoj udaljenosti od kompjutora, možda nekoliko metara, ali i neko-

° Prevodilac se ovdje ne može suspregnuti da ne posumnje u ovu potrebu, barem što se tiče jugoslavenskog korisnika. Stoga neke dijelove ovog poglavlja treba promatrati kao čisto teorijsku raspravu.



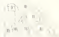
Neke od postojećih naprava (periferije) koje se mogu priključiti preko porte na kompjutor (1): tastatura (2) ima svoju vlastitu portu, kao što je ima televizor (3) koji može služiti kao displej i jedinica magnetskog diska (disk drive) (5). Drugi su uređaji priključeni preko standardne serijske ili paralelne porte adaptirani elektronički pisaći stroj (4), pisač ili printer (6), kazetofon (7) i roller ball, naprava za vođenje kursora po ekranu (8).

liko stotina kilometara, treba biti siguran u točnost podataka koji dolaze ili se primaju. Tada paritetni bit (parity bit) opisan na stranicama 12-13 dolazi do izražaja, kao i još neki sofisticiraniji načini provjere ispravnosti prijenosa podataka. Kompjutorska komunikacija vrlo je značajno područje i knjige koje su o tome napisane zauzimaju cijele police u knjižarama.

Unutar kompjutora, procesor shvaća svaku portu kao dvije memorijske lokacije: jednu za podatke koje treba odaslati ili primiti, a drugu koja će mu reći na koji se način taj proces odvija pomoću podataka ili statusnih bajtova. To procesoru olakšava život. Ako šalje podatke preko porte, on će samo od vremena do vremena pogledati na statusni bajt da bi saznao treba li još podataka ili ne. Ako treba, tekući program se zaustavlja i procesor šalje nove podatke porti. Ali kad izvana primi signal da više ne treba podataka, statusni bajt se mijenja i procesor prestaje s predajom, te nastavlja posao kojim se ranije bavio.

Postoje dva načina da se razluči podjela zadataka: prvi se sastoji u primjeni postupka 'interrupt' (obustava). Procesor prima signal svaki put kada u statusnom bajtu jedan bit promijeni vrijednost od 0 na 1 tražeći dalje podatke. Taj znak mu kazuje da treba odgoditi ono što upravo radi i skočiti na neki drugi program. U slučaju koji nam služi za primjer, signal javlja da treba slati podatke na portu. Kada je to izvršeno, interupt se prekida i procesor se vraća programu koji je prekinuo. Interupti dakako mogu služiti i u mnoge druge svrhe.

Drugi način je tzv. polling (poll-lista, spisak). Sastoji se u tome da procesor od vremena do vremena pogleda na portu da li tamo postoji neki novi zahtjev za podacima. On može tako optrčati, prolistati sve na njega priključene periferije, pitajući svaku posebno da li joj štogod treba. U to je ispitivanje dakako uključena i tastatura. Najvještiji konsnik tipkat će oko deset znakova u sekundi, što znači da će razmaci između dviju tipki trajati 1/10 sekunde, a to je procesoru i previše da bi optrčao sve periferije (uključivši i tastaturu), te svakoj posvetio dužnu pažnju. Budući da on sve to obavlja frekvencijom od nekoliko milijuna titraja u sekundi, ostaje mu još dovoljno vremena da se bavi ostalim poslovima.



Još nekoliko naprava koje se mogu priključiti na kompjutor preko porte. Telefonski modem (9), displej sa tastaturom (10), ploter ili jedinica za crtanje (11), grafički tablet (12), robot (13), palica (joystick) i dugme (knob) (14) i na kraju miš (15). Pomičući miša po stolu, kuglica ispod njega se okreće i vodi kursor po ekranu.

EKRANI

Koliko god bude trajao vijek ove knjige kao kompjutorski displej služit će katodne cijev kao i u svakom televizoru Vjerojatno ce u buducnosti njenu ulogu preuzeti LCD displej ali o tome kasnije Katodna cijev je iznenađujuće složena naprava i da nema tako ogromne popularnosti televizije njena proizvodnja bila bi beznadno skupa

Katodne cijev je građena poput neke velike boce s ravnim dnom ekranom koje je prevučeno tankim slojem fosforne soli U njoj vlada vakuum a u grlicu se nalazi elektronski top koji ispaljuje tanahnu zraku elektrona u smjeru ekrana Na mjestu gdje zraka pogađa ekran pojavljuje se svijetla točka Ako se dovede odgovarajuci napon na dva para otklonskih pločica zrakom se može upravljati i voditi je u svim smjerovima Na taj način moze se crtati po ekranu Svijetla točkica putuje ekranom ogromnom brzinom od nekih 40 000 kilometara ne sat pa zato samo u jednoj sekundi projun više puta cijelom površinom ekrana Budući da fosfor ima svojstvo da svijetli još nekoliko tisućinki sekunde nakon što je zraka prošla stvara se iluzija stalnog osvjetljaja bez treperenja

Postoje dva načina izazivanja slike na ekranu Prvi se naziva vektorskim Zraka elektrona vodi se po ekranu poput olovke u ruci i tako iscrtava oblike To daje vrlo kvalitetnu sliku ali radi vrlo sporo

Drugi način kakav je gotovo u svih kompjutora, naziva se rasterskim skaniranjem Zraka elektrona ispisuje ogromnom brzinom gusti raster horizontalnih linija Svijetla točka jureci svojim putem pali se i gasi te na taj način iscrtava sliku sastavljenu od niza točkica koje su tako blizu jedna drugoj i tako sitne da ih oko vidi kao neprekinute linije i fine tonske prelaze

Na suvremenom televizijskom ekranu ima 600 × 600 točaka što čini ukupno 360 000 A znamo da čak i tako gusti raspored točkica ne daje osobito oštru sliku Ako svaku točku pro-

Unutrašnjost katodne cijevi Zraka elektrona (crvena) ispaljena iz elektronskog topa najprije prolazi kroz kolimator koji je fokusira Tada nailazi na dva para otklonskih pločica čija je električka polja skreću lijevo ili desno gore ili dolje Dalje putuje kroz vakuum sve dok ne pogodi unutrašnju stranu ekrana koja je presvučena fosforom i na njoj ostavi trag u vidu svijetle točke Top se može uključiti i isključiti a zraka upraviti na bilo koje mjesto ekrana i njome naslikati ili napisati neko slovo ili broj Elektronska zraka mora ispisati sliku svaka 1/25 sekunde Pri vektorskom skaniranju zraka iscrtava sliku poput olovke vođene rukom i daje vrlo kvalitetnu sliku Pri rasterskom skaniranju putuje u paralelnim horizontalnim linijama gasi se i pali kada je potrebno Ovaj drugi način je brži ali daje ponešto grubu sliku s manje detalja Najveći dio mikro kompjutora služi se tim načinom

fosforni ekran

elektronski top

kolimator

horizontalne otklonske pločice

TV slika



vektorsko skaniranje



rastersko skaniranje



metramo kao zasebnu jedinicu (jednica se naziva 'pixel', što je kovanica od riječi 'picture' i 'cell' — ćelija slike) — treba najmanje po jedan bajt za svaku točku, pogotovo kad se radi o slici u boji. To iznosi pola megabajta RAMa, i to samo za kontrolu ekrana, što je gotovo nemoguće postići sa 8-bitnim kompjutorima, a čak i kod 16-bitnih zauzeti će preveliki dio memorije.

Kod 8-bitnih kompjutora slika je podijeljena na manji broj, ali zato većih piksela. Standardna organizacija ekrana iznosi po 80 znakova raspoređenih u 24 horizontalna reda. U svaki od tih 1920 kvadratića kompjutor upisuje jedan od unaprijed fiksiranih znakova, šifriranih svojom ASCII šifrom (vidi str. 12-13). Kako za svaki ASCII znak treba samo po 1 bajt, za kontrolu slike otpada 2 kilobajta RAMa.

ASCII znak za A je 65 ili bajt '01000001', ali to dakako još nije dosta da bi se na ekranu naslikalo slovo A. Problem je riješen tako da je u ROM memoriji pohranjen oblik slova A u matrici koja se obično sastoji od 9 redova po 5 točaka. Kada treba naslikati A na ekranu, kompjutor zna da u prvom gornjem redu treba svijetliti samo srednja točka, dok su ostale ugašene. U



slijedećem redu mora svijetliti svaka druga, a u onom ispod njega, prva i zadnja, i tako do kraja zadnjeg reda (vidi sliku lijevo). Neki kompjutori rade s matricom u visini od 7 točaka. Kada se radi o malim slovima, kao što su npr. 'y' i 'p', ona se ne mogu prikazati s onim dijelovima koji vise ispod donjeg ruba retka, što je dosta nespretno.

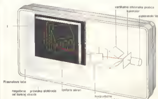
Sasvim druga vrst ekrana je ona izrađena u tzv. 'solid state' tehnologiji°. Na njima se slika formira pomoću LED°° dioda ili tekućih kristala. Na diodnom displeju svaka točka nastaje od sićušnog izvora svjetlosti, a na displeju od tekućih kristala (LCD)°°° točke nastaju kad se struja propusti kroz elektrode, kako je opisano uz sliku desno.

U osnovi takvi bi ekrani morati biti mnogo bolji od klasičnih katodnih cijevi. Ne samo bolji, već i manji, jeftiniji i ekonomičniji u potrošnji energije. Međutim, da bi se postigla dovoljno visoka rezolucija (oštrina), treba toliko mnogo točaka i isto toliko elemenata koji će svaku kontrolirati, da je praktička izvedba još uvijek opterećena premnogim tehničkim i tehnološkim zaprekama.

° Pojam 'solid state' tehnologije odnosi se na primjenu čvrstih materijala (tranzistora, diode, feritnih jezgri itd.) u izvedbi elektronskih uređaja. Prim. prev.

°° LED — akronim za Light Emitting Diode — svjetleća dioda. Prim. prev.

°°° LCD — akronim za Liquid Crystal Display — displej od tekućih kristala. Prim. prev.



Plosnati ekran

Postojeće katodne cijevi nespretne su zbog svoje velike zapremine. Nekoliko proizvođača radi na projektu nove plosnate cijevi kod koje se top ne nalazi iza ekrana, nego sa strane. Elektroni tada moraju putovati kompliciranim putem i teško je izbjeći distorziju i neoštrinu slike.



Sinclairov prijedlog za dizajn prenosnog televizora s plosnatim ekranom. Slika na ekranu je dakako umontirana fotografija.

LCD displej
(Liquid Crystal Display)

Dolje lijevo: Ne postoji neki ozbiljan razlog da se displej od tekućih kristala ne bi mogao upotrijebiti za TV sliku ili kompjutorski monitor. Postoje samo neke poteškoće oko stabilnosti, osjetljivosti na temperaturu i velikog broja logičkih elemenata koji bi upravljali slikom od 600 × 600 točaka.

Dolje desno: LCD displej na ručnim satovima sastoje se od dviju staklenih ploča između kojih se nalazi tanki i providni sloj organske tekućine. Segmenti brojeva nalaze se ispred stakla u vidu transparentnih elektroda. Kada se na



njih dovede napon, tekućina ispod njih potamni. Dolje je prikazano kako nastaje broj 5.





Napisati DOBRO JUTRO (GOOD MORNING) točno na sredini ekrana (24 retka po 40 znakova) nije sasvim jednostavno. Ispravno pozicioniranje zahtijeva ponešto računanja, što se dakako može riješiti i programski. Najprije treba kursor postaviti HOME (kući) — u gornji lijevi kut ekrana. Nakon toga treba ga spustiti 12 redaka niže da dođe do središnjeg retka. Fraza GOOD MORNING ima 11 slova i jedan razmak — treba dakle otipkati (40-12) 2 = 14 razmaka da bi kursor došao na poziciju početka. Sada možete otipkati svoju jutarnju po ruku koja će biti točno kako ste htjeli na sredini ekrana



Saznali smo kako slike nastaju na ekranu. Sad je pitanje kakve su to slike?

Očito je da slova i brojevi spadaju u najvažniju grupu slika. Jeftiniji kompjutori koji se služe običnim televizorom kao monitorom imaju na ekranu po 40 piksela u jednom horizontalnom redu. Oni skuplji s posebnim monitorom, budući da imaju kvalitetniju sliku, obično rade s 80 znakova, pa su i sitnija slova dovoljno čitljiva.

Svako slovo ispunjava prostor jednake, standardne veličine, bilo da se radi o 40 ili 80 znakova (to znači da isto kao i na pisaćem stroju oko slova i ima mnogo više praznog prostora nego oko slova w ili m). Kompjutori bolje konstrukcije imat će mala i velika slova, a slova y, j, p, q i g koja prelaze ispod donjeg ruba reda bit će prikazana na pravilan način. Kako se većina kompjuteraša u svijetu služi engleskim jezikom, slogovi slova na kompjutorima su ili engleske ili sjevernoameričke jezične varijante. Razlika je samo u znakovima # i £ (funta) za koje vrijedi ista ASCII šifra. Danas je evropsko tržište postalo tako značajno da su proizvođačima isplati dodati posebne znakove, kao što su npr. akcentuirani francuski i preglašeni njemački vokali. To moraju biti posebna slova, budući da se na ekranu ne mogu (kao na pisaćem stroju) kombinirati dva znaka u jedan. Zbog toga u nekim kompjutorima postoji poseban jezični preklopnik kojim se uključuje odgovarajući dio ROMa predviđen za takve jezične posebnosti.

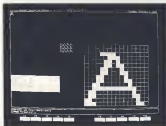
Ako želite napisati veće slovo od normalnoga, morate ga sagraditi ovako

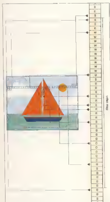
```
PPPPPPPPPPPP
 PPP       PP
 PPP         PP
 PPP       PP
 PPPPPPPPPP
 PPP
 PPP
 PPP
 PPP
PPPPP
```

Već se pojavljuju kompjutori koji mogu pisati u dvjema veličinama, većim slovima za naslove, a manjim za obični tekst. Tako bi se sa 16-bitnim procesorima s mnogo slobodnog memorijskog prostora mogla ispisivati slova proporcionalnom metodom koja slovima različite širine daje onoliko prostora koliko svakom zaista treba, poput slova u ovoj knjizi.

Kompjutor može nacrtati svaki oblik koji se može izraziti točkastom strukturom ekrana. To znači da ni arapsko pismo, na primjer, ne predstavlja poteškoću, jer njegovi oblici nisu nimalo složeniji od oblika abecede. Arapski ima manje slova od latinice i sva mogu stati u ROM i biti adresirana sa po jednim bajtom kao i sve ostale ASCII šifre. S japanskim je mnogo teže. Katakana, jednostavnija od dviju varijanti japanskog pisma, ima oko 2 000 znakova, od kojih su mnogi složeniji od bilo kojega znaka abecede. Ipak se i to može riješiti ako se svakom znaku da

Neki 16-bitni kompjutori omogućuju korisniku da sam dizajnira svoja slova. Umjesto da su permanentno spremljeni u ROMu (kao na 8-bitnim računalima), čuvaju se na disku i očitavaju svaki put iznova kada se kompjutor uključi. Ovdje vidimo program za dizajniranje slova A na računalu CORVUS CONCEPT





Lijevo, kako kompjutor reproducira slike u boji. Ekran je (obično) podijeljen na 24 reda po 40 znakova u svakom redu, što iznosi 960 piksela. Dio memorije koji se zove VIDEO MAP na svaki red troši po 40 bajtova raspoložive memorije. Svaki bajt RAMa ima broj šifre koja će generirati željenu boju u odgovarajućoj ćeliji. Na slici jedinice u jednom redu ima bijele (W), plave (B), crvene (R) i narančaste (O) boje. Program koji se vrti negdje u RAMu poslat će u VIDEO MAP šifru svake boje za svaku ćeliju. Poseban VIDEO sklop čita redom šifre i generira odgovarajuću boju na ekranu.

Treba znati da se ekran svaki put iznova iscrtava i to ima dvije posljedice. Prva se odnosi na osnovni princip animacije, a taj je da se slika mijenja samo u detaljima koji se pomiču. Na primjer, ako želimo prikazati Paju Patka koji govori, slika ostaje nepromijenjena osim njegova klepetava kljuna. Nova slika se iscrtava svake $1/25$ sekunde i kljun je svaki put u malo drugačijem položaju jednako kao i u filmskoj animaciji, samo s tom razlikom što se slika crta na ekranu umjesto na celuloidnoj foliji. Druga se odnosi na kompjutor kojemu treba dati dosta vremena za proračun svih promjena i iscrtavanje cijele slike, a sve to u jednoj dvadesetpetinki sekunde. Ako znamo da čak i jedan displej slabije kvalitete mora imati oko 8KB za video map (a to je dosta drastično ograničenje), promjene moraju biti polagane ili treba raditi na snažnijem kompjutoru. Treća je mogućnost snimati sliku po sliku što se primijenjuje i u filmskom snimanju uz pomoć kompjutora. Pri tome nije važna brzina jer će se film ionako projicirati standardnom brzinom.

Mnoge stvari se ipak mogu izvesti jednostavnom dvodimenzionalnom kompjutorskom animacijom. Glavna poteškoća je u stalnom i neprestanom izračunavanju novih podataka koje treba mijenjati u video mapu (updating), pa je prava trodimenzionalna animacija izvediva samo na vrlo snažnim kompjutorima (vidi stranice 114-115).

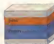
Ipak ima nekih trikova pomoću kojih se na kućnim kompjutorima mogu izvesti dobra dvodimenzionalna animacije. Postoje grafički taveći prostor na ekranu i po dva bajta za adresiranje (s jednim bajtom može se adresirati samo 256 znakova).

Osim slova i brojeva koji se nalaze u ROMu mnogi amaterski kompjutori opremljeni su još i posebnim setom raznih grafičkih znakova, velikih poput slova, od kojih se mogu sastavljati jednostavne slike i koristiti u kompjutorskim igrama.

U svim grafičkim prikazima ekran se iznova iscrtava svake $1/25$ sekunde. Za taj posao rezerviran je posebni prostor u memoriji koji može biti i dio glavnog RAMa. Taj se dio memorije zove VIDEO MAP. Svaki bajt (ili dva bajta) odgovara jednom pikselu (vidi stranu 30-31). Na suvremenom kućnom kompjutoru rezolucija ekrana često dostiže do 200 × 400 piksela, tako da video map mora raditi s nekih 80 000 piksela. Ako je slika jednobojna, svaki piksel je označen s jednim bajtom, što iznosi 10K RAMa. Za četverobojnu sliku (crveno, zeleno, plavo i crno) treba po dva bajta (dva na kvadrat je četiri), što iznosi 20K RAMa. Sve što procesor unese u video map istodobno se prikazuje na ekranu i to frekvencijom od 25 slika u sekundi, pa se zato kao i na crtanom filmu mogu izvoditi jednostavnije i polaganije animacije.

Grafika u visokoj rezoluciji svejedno na kojem računalu uvijek zauzima veći memorijski prostor od grafike u niskoj rezoluciji. Zbog toga na malim računalima ostaje malo memorijskog prostora u RAMu kada se vrte grafički programi u visokoj rezoluciji.





niska rezolucija





visoka rezolucija

bleti visoke rezolucije koji rade na ponešto drugačiji način, ali daju zadovoljavajuće rezultate.

U jednostavnijim računalima obično ima uz standardna slova i brojeve još i određan broj grafičkih znakova koji se mogu postaviti na bilo koju točku ekrana, a zauzimaju isti prostor kao slova. To su obično svijetli ili tamni kvadrati ili pravokutnici u različitim kombinacijama. Od njih se uz malo strpljivosti mogu sastaviti jednostavne slike.

Grafički programski jezik, kao na primjer LOGO (str. 77), sadrži neke grafičke naredbe kao dodatak postojećem BASICu (strana 58-61). To su naredbe kojima se može pozicionirati kursor na bilo kojem mjestu ekrana, povući crtu između dviju točaka, nacrtati kružnicu ili ispuniti neki prostor bojom. Isti zadaci mogu se dakako rješavati i pomoću posebnog softvera.

Na kompjutoru koji je predviđen da radi u boji može se definirati 16 odnosno 256 boja, svaka pod svojim brojem, a obično se određuju na početku programa.

Pretpostavimo da u nekoj animiranoj sceni sva lica imaju boju broj 6. Odjednom se pojavljuje strašni duh i svi posive od straha. Ovo se izvodi promjenom boje 6 u, recimo, boju 8 — sivu.

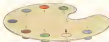
Postoje dva načina animacije. Prvi se naziva 'PAGING' i ima dva ili više video mapa. Slika se uzima iz prvoga, za što treba obično više od nominalne 1/25 sekunde, a zatim se sljedeća uzima iz drugoga. Za to vrijeme se u prvom priprema treća slika. To dobro funkcionira, ali je dakako skuplje, jer zahtijeva više prostora u RAMu.

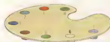
Drugi bismo mogli nazvati (prema terminologiji crtanog filma) 'multiplan' sistemom, sličnim onome kojim se služe filmski animatori kada sliku razlože po dubini, na planove. Hardverski je rezerviran prostor u memoriji gdje se spremaju samo dijelovi slike, koji se mogu postaviti na bilo koje mjesto ekrana. Kao primjer neka posluži scena u kojoj vidimo dječaka što hoda po ledini na kojoj je stablo, a na pozadini plavog neba su oblaci. Sliku treba najprije razložiti na planove. Planovi se numeriraju od 0 na više, s tim da je nulti plan onaj najdalji od gledaoca. Sada će se na planu 0 naslikati nebo i zelena ledina, na planu 1 stablo s krošnjom, na planu 2 glava i gornji dio tijela, na planu 3, 4 i 5 noge u tri faze pokreta. Kako neki sistemi za kompjutorsku animaciju rade samo s jednobojnim planovima, tako će se, na primjer, plan 1 morati razložiti na dva plana, jedan sa smeđim stablom, a drugi sa zelenom krošnjom. Pretpostavimo da radimo sa savršenijim sistemom koji podnosi više boja na svakom planu. Kako sadržaj svakog plana pokriva dijelove prethodnog, snimanje će se vršiti ovim redom: plan 0, plan 1, plan 2, 3, 4 i na koncu 5. Planovi 3, 4 i 5 ponovit će se naizmjence nekoliko puta da bi se postigla iluzija hodanja, dok će se istodobno planovi 0 i 1 kretati u suprotnom smjeru, ali različitim brzinama. Tako će iluzija da se dječak kreće kroz pejzaž biti potpuna.

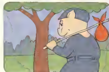
Sve se ovo može nacrtati na nekoliko načina. Može se crtati upravljajući kursorom pomoću palice (joystick) i — ako je bolje upotrijebiti grafički tablet (vidi str. 110-111). Pojedini dijelovi slike mogu se unaprijed nacrtati i spremiti na disk ili magnetsku vrpcu, pa kasnije spretnim kombiniranjem pojednostavniti posao.

Postupak se može primijeniti i u drugim slučajevima. Na primjer, u nekom industrijskom postrojenju potrebno je u isto vrijeme kontrolirati tri radna procesa sa tri različita instrumenta. Kompjutorskom animacijom mogu se konvencionalni instrumenti zamijeniti ekranom. Planovi će se u tom slučaju organizirati na sljedeći način: plan 0 programiran je za kazaljku koja se kreće horizontalno. Plan 1 prikazuje kazaljku

Savršeniji način bojenja slike od onog sa strana 33: u RAMu se nalazi prostor za šifre boja koje se odabiru s jedne palete i unose u video map. To je način da se boja na nekoj površini zamijeni drugom. Na slici vidimo prasčića obojenog bojom 6 — ružičastom, a kada ugleda duha, iznenada posivi.









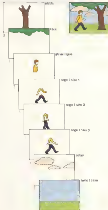
Premda se tipkama tastature kursor može lako voditi po ekranu, ipak to nije najlakši način da se prave crteži. Pravo rješenje je grafički tablet po kojem se crta elektronskim pisalom koje emitira visokofrekventne radiosignale. Njih detektira vrlo fina žičana mrežica utaljena u površinu tableta. Elektronika u tabletu konvertira ih u X i Y koordinate, šalje kompjutoru, a ovaj ih reproducira na ekranu.



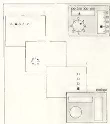
koja se kružno kreće, a plan 2 kazaljku koja se kreće vertikalno. Kompjutor dobija podatke od senzora i kazaljke 0 i 2 kreću se po x i y koordinatama, a kazaljka 1 tako da se x i y koordinate preračunavaju i padaju na obod kružnice. Ostaje još samo da se nacrta podloga s oznakama vrijednosti na pojedinim skalama, koja ovaj put stoji na planu 3.

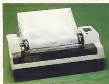
Takva animacija može dati samo vrlo jednostavne rezultate. Slike mogu biti samo dvodimenzionalne i ne mogu se rotirati. Pravi sistemi za animaciju moraju biti predviđeni tako da mnoge složene operacije izvode automatski (vidi str. 112-115).

Razlaganje slike na planove za svrhu animacije neki kompjutori (Atari, na primjer) imaju nekoliko memorijskih prostora u koje se može spremiti samo dio neke slike, a kasnije se može pozicionirati na bilo koje mjesto ekrana. Planovi s višim brojem prekrivaju one s nižim, i tako se može naslikati dječaka koji hoda iza stabla, a ispred oblaka i zelene livade. Izmjenom triju verzija nogu i ruku i pomicanjem pozadine u horizontalnom smjeru crtež može oživjeti.



Sasvim desno, sustav planova može se primijeniti i za druge svrhe: ovdje je nacrtan instrument s tri kazaljke. Kompjutor prevodi rezultate triju različitih mjerenja u slike pomičnih kazaljki na brojčanicima.





ABCDE
ABCDE

Povećani otisak zvjezdice (daisywheel) — gore — i točkaste matrice (dot matrix) — dolje
Desno princip Olivettijevog mlaznog printera U staklenoj cjevčici nalazi se čvrsta električki vodljiva boja Uski vrh cjevčice je blizu papira a uz njega elektroda čiji je drugi kraj umočen u boju Kada treba ispaliti kapljicu boje na papir struja visokog napona izazove iskru i ze jedno s njom izleti nešto rastaljene boje na papir

Zvjezdica (dolje desno) se svojim poput cvijeta krhkim laticama i finim elegantnim slovima jedna je od ljepših stvari u kompjutorskom svijetu (daisy — engl tratinčica)
Dolje je makro snimak slova na zvjezdici tvrtke Qume

Često zaželimo da neku sliku ili tekst prenesemo s ekrana na papir i tako je sačuvamo Upravo je zbog toga izmišljen printer ili pisac Između njega i električnog pisaćeg stroja postoje mnoge sličnosti ali i razlike Gumeni valjak vodi papir a slova se kao i na pisaćem stroju otiskuju preko vrpce samo što nisu na dugim metalnim polugama jer takav mehanizam ne bi bio podesan za pisanje velikom brzinom poput one kojom radi printer

Postoje dva načina otiskivanja slova Prvi se služi zvjezdicom (daisywheel) Ispupčena slova nalaze se na krajevima latica metalnog ili plastičnog koluta (desno) Kada slovo dođe na predviđeno mjesto udari ga mali čekić i preko vrpce s bojom otiskuje na papir

Druga metoda naziva se matričnom Slova se formiraju na papiru na isti način kao i na ekranu — nizom točkica U glavi printera nalazi se okomiti red iglica i sa svakom posebno može se udariti po vrpci i ostaviti otisak na papiru (U nekim novijim izvedbama točkice se ispisuju mlazom boje — vidi dolje)

Printeri sa zvjezdicom općenito su skupi bučniji i polaganiji ali njihov otisak je čist i vrlo lijep Savršenije izvedbe mogu pisati proporcionalno razmaknuta slova — kao što su slova u ovoj knjizi i zauzima manje prostora nego m

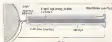
Matrični printeri su brzi i tiši ali je njihov otisak grublji i nečistiji Ali ako su im iglice dovoljno sitne i postavljene dosta blizu, otisak će biti gotovo jednak onom na dobrom pisaćem stroju Danas su već ti printeri toliko usavršeni da će po svoj prilici uskoro sasvim istisnuti one sa zvjezdicom

Matričnim printerom može se otisnuti bilo koji znak ili oblik u nekoliko različitih veličina i u istom redu Ako je softverom predviđeno može se prelaziti s jednog pisma na drugo sa latinice na ćirilicu na rusko japansko ili arapsko pismo i ponovo na latinicu

Obe vrsti mogu jednostavno preći u grafički način rada (grafic mode) otiskujući samo po jednu točkicu a valjak i glava se pomiču u tako malim koracima da se točkice preklapaju iscrtavajući jednolične crte i površine Mogu se također praviti grafike u nekoliko polutonova jednostavnim povećanjem razmaka između točkica što djeluje poput rastera u tiskarskoj tehnici

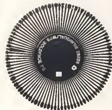
Najveći nedostatak opisanih printera je vrlo složena mehanika podložna velikim naprezanjima pa stoga i kvarovima Da bi se tome doskočilo u nizu istraživanja koja se ubrzano provode na tržištu su se već pojavila dva nova rješenja Postoji već mlazni printer koji u pravom smislu riječi ispaljuje sićušne mlazove boje po papiru Zasada je najelegantnije rješenje Olivettijev mlazni printer koji pomoću električne iskre sitnim kapljicama boje ispisuje znakove To se odvija vrlo brzo i sasvim tiho a budući da glava za pisanje i papir nemaju nikakvog fizičkog kontakta ništa se ne može ni istrošiti osim kapsule s bojom koja se uostalom mijenja vrlo jednostavno

Najsavršenija ali i najskuplja verzije printera koristi lasersku zraku koja velikom brzinom ispisuje tekst I ovdje nema mehaničkog kontakta s



papirom pa je to najbrži način ispisivanja teksta koji se fiksira na papiru nekom vrsti kserografskog postupka

Sve bi to bilo vrlo lijepo kada nedostatak opće prihvaćenih standarda ne bi siromašnom korisniku zagorčavao život Pretpostavimo da ste pisac i želite pisati roman pomoću kompjutora i printera Ili jednostavno želite pisati pisma s lijepim i urednim slovima i sa izravnanom desnom i lijevom marginom (baš kao u ovoj knjizi) Nabavit ćete sav hardver i softver koji vam treba i onda slijedi razočarenje Ustanovljujete da je nemoguće sve to povezati i natjerati da poslušno radi Problem je u tome što treba izraču-



Najbrža i najjeftinija vrst printera je ona s točkastom matricom. U glavi za pisanje nalazi se 7 ili 9 iglica u redu, jedna iznad druge. Svaka može udariti po vrpci kakva se upotrebljava na pisaćim strojevima i otisnuti točkicu na papiru. Upravljane kompjutorom, iglice u raznim kombinacijama otiskuju slova i brojeve. Na slici desno prikazano je povećano slovo A otisnuto matricom 7 × 5 točaka. Neki printeri imaju dva reda iglica postavljenih sa pomakom od pola promjera točkice. Na taj način se točkice preklapaju, pa su slova sličnija otisku pisaćeg stroja (desno). Isti efekt se postiže kada se dva puta pređe preko istog reda. Takvim printerom mogu se iscrtavati i grafike. Glava s iglicama putuje s jednog kraja papira na drugi, a posebni motor povlači papir.

papir
vrpca s bojom
zavojnica magneta (jedna za svaku iglu)
iglice od volframa
glava printera
upravljački sklop
ROM memorija
šifra slova

Kompjutor šalje printeru signal da otipka slovo N. Procesor printera prevodi sliku tog slova spremljenu u ROMu u odgovarajući raspored iglica u glavi za pisanje.

zvjezdica
vrpca s bojom
papir
valjak
stepenasti motor za okretanje valjka
klin
kolica
motor za povlačenje vrpce
servo motor za okretanje zvjezdice
čekić
nazubljeni remen
servo motor za pogon kolica

Druga vrst printera slična je električnom pisaćem stroju. Služi se slovima organiziranim po obodu zvjezdice. Da bi se otisnulo neko slovo, procesor u printeru okrenut će zvjezdicu, postaviti pravo slovo na pravo mjesto. Po njemu će tada udariti klin, a slovo će preko vrpce s bojom ostaviti otisak na papiru. Otisak je čišći nego u matričnog printera (isti je kao u električnog pisaćeg stroja), ali je printer sa zvjezdicom osjetno polaganiji i skuplji.

nati koliko riječi može stati u jedan red pa rastaviti zadnju riječ i prebaciti je u slijedeći red. Da bi se ispunio cijeli red i posljednje slovo posljednje riječi postavilo točno na desnu marginu treba ubaciti dodatne razmake između riječi. Kompjutor će doduše sve to lako izračunati ali ako su printer i softver od različitih proizvođača neće se potpuno razumjeti i na vama je da ih uskladite, što znamo može biti vrlo komplicirano.

Područja kompjutora nalazi se danas u razvojnom procesu s mnogo sjajnih ideja ali bez ikakve povezanosti. Često je za korisnika najveći problem upravo genijalnost konstruktora. U principu kompjutor može sve i ono što je važno i ono što nije. Tjerati ga da radi nevažne stvari jedva je vrijedno uložena truda. Bolje je zadovoljiti se skromnijim rješenjem ali zato korisnim nego težiti nedostižnoj perfekciji.

Kompjutorski fotoslog

Jedno je stranica ispisana printerom a posve drugo ona otisnuta pravom tiskarskom tehnikom.

U tiskarstvu su naime ustanovljeni mnogo viši standardi jednoličnosti i točnosti pisma. Slova su preciznija pa zato mogu biti i gušće raspoređena. U novinama je, na primjer, tekst mnogo gušći nego na pisaćem stroju a slovoslagar ima na raspolaganju mnogo veći izbor slova i razmaka između slova i redova. U stvari slaganje slova je tako stručan posao s tako velikim mogućnostima da zahtijeva vrlo usku profesionalnu specijalnost. Samo pravi profesionalni slovoslagar može obaviti sve one posebne zadatke kojih postajemo svjesni tek kada na stranici što je čitamo nešto nije u redu. Običan kompjutor nije tome dorastao pa zato te poslova obavlja poseban, kompjutorom vođeni fotoslog.

Postoje tri vrsti kompjutorskog fotosloga. Prva se služi staklenim diskom s fotografskim negativom slova i simbola različitih oblika i veličina. Disk se nalazi ispred foto-papira a slova se okretanjem diska postavljaju na određeno mjesto osvjetljavaju se i uobičajenim fotografskim postupkom razvijaju. Nastaje slika' teksta kakva će biti otisnuta u knjizi.

U drugoj vrsti slova se generiraju na ekranu katodne cijevi vrlo visoke rezolucije. Složeni tekst se i u ovom slučaju snima standardnom fotografskom tehnikom. Treći način se služi laserskom zrakom koja kontrolirana softverom ispisuje slova na fotoosjetljivoj podlozi.

Veliki problem je rastavljanje dugačkih riječi na prijelazu u drugi red. Postoje programi koji to obavljaju automatski ali redovito ne funkcioniraju dobro. Mnogo je važniji vješt radnik nego vrst stroja jer znamo da nema pouzdane zamjene za dobra slovoslagara.

U posljednje vrijeme razvija se jedan novi postupak pri kojem već u pisanju teksta uz pomoć kompjutora sâm autor ubacuje kontrolne znakove (koji će služiti određivanju definitivnog izgleda tiskanog sloga) pa se onda direktno ulazi u kompjutor fotosloga. Na taj način otpada ponovno tipkanje teksta u tiskari što doduše predstavlja veliku uštedu vremena ali nije sigurno hoće li tiskarski sindikati dočekati takvu novotariju s oduševljenjem.

I sasvim jeftini mlazni ili printeri s višebojnom vrpcom mogu izvesti ovakve elegantne šarobojne grafike. Ova predstavlja grafički prikaz razvedenosti zemljine površine.

Ploteri

Na strani 36 opisano je kako se printerom može dobiti slika na papiru. Kako su takve grafike dosta grube, napravljene su posebne naprave, ploteri, koji poput pravog crtača izrađuju crteže perom.

U osnovi radi se o peru što ga dva motora posebne izvedbe (step motor) pokreću stepenasto malim koracima za vertikalni i horizontalni smjer. Koraci mogu biti posve mali, od djelića milimetra, pa sve do raspona cijelog plotera. Ako motor radi samo za horizontalni smjer, pero će povlačiti horizontalnu liniju. Vertikalna linija će nastati ako motor radi samo za vertikalni smjer. Ako pak rade oba motora naizmjence istom veličinom koraka, linija će se penjati (ili spuštati) pod kutom od 45 stupnjeva. Ako su koraci dovoljno mali (oko 1/10 debljine pera), na liniji se neće primijetiti stepenasti način kretanja, bit će ravna i glatka kao da je iscrtane ravnalom.

Što je ploter skuplji, to su i crteži koje izrađuje savršeniji. Dobar ploter vrlo je skup. Ima ih koji rade s nekoliko pera u različitim bojama, pa je zabavno promatrati crtaču glavu kako užurbano uzima pero sa zelenom bojom, povlači nekoliko zelenih crta, zatim uzima drugo i dodaje nekoliko crvenih, a sve sa savršenom preciznošću i brzinom.

Budući da ovakvi ploteri izrađuju nacrte za vrlo skupe projekte, cijene im nije važna. Ako su instrukcije za ploter pohranjene u središnjoj bazi podataka, nije teško koordinirati cijelu konstruktorsku grupu i uštedjeti rad gomile strpljivih crtača.

Konstruktori plotera nalaze se pred teškim problemima, jer je svrha tog stroja da izrađuje crteže većom brzinom i savršenijom preciznošću od profesionalnog crtača. Kako spriječiti pero koje je čekalo jednu ili dvije milisekunde i onda naglo pojurilo na početak nove linije, da ne zadrhti ili da ne ode predaleko. Borba s praznim hodom u ležajevima i vodilicama također je mučna, jer ako su preveliki, pero se neće vratiti na točku s koje je krenulo, a ako su pak premaleni — još je lošije.

Glave plotera su pri naglom ubrzavanju i zaustavljanju izložene vrlo velikim naprezanjima. Zbog toga oni najveći, plosnati ploteri (flat bed) imaju kablove debele poput prsta i dok rade moraju biti pokriveni staklenim pokrovom, jer bi neki neoprezniaković mogao ostati bez prstiju ako bi ih nesmotreno približio crtačoj glavi što leti upravo nesmiljenom brzinom.

U svaki nacrt treba ucrtati mnoštvo slova i natpisa, standardnih likova, elipsa, kvadrata i kružnica. Stoga ploterov softver koji upravlja crtačom glavom, ima i nekih prednosti koje ima i printer. Da bi se ucrtao neki naslov na nacrtu, ne treba šablonom ispisivati slova. Dovoljno je poslati preko kompjutora naredbu "PRINT "Zupčanik 25mm"". Želimo li kružnicu, onda "CIRCLE 2.6, 3.56, 6.1". (Brojevi u ovoj poruci označavaju koordinate središta i promjera kružnice.)

Ploter nije ništa pametniji od printera, pa i za njega treba posebni, koji put i skupocjeni softver. Na primjer, softver za izradu trodimenzionalnih nacrta nije nimalo jednostavan i, dakako, ni jeftin, a nisu ni zahtjevi koji se postavljaju hardveru skromni. Da biste stekli kakav-takav pojam o složenosti tih problema, treba pročitati što piše o tome na stranama 102 i 103.

Gore: ima plotera koji rade tako velikom brzinom da neopreznom operatoru mogu ozlijediti ruku.

Sredina: četvorobojni ploter Sharpova džepnog kompjutora nema nekih velikih mogućnosti, ali je ipak pravo malo čudo tehnike.

Dolje: kompjutorski perivoj Scotta Browningga i Turnera. Pretpostavlja se da je to prvo umjetničko djelo stvoreno kompjutorom i prikazano na Kraljevskoj akademiji. Zamislimo što bi Louis XIV učinio u Versaillesu da je imao kompjutor.

MAGNETSKA MEMORIJA



Tipični uredski kompjutorski sustav s dvostrukim diskom (disk drive). U rukama djevojka vide se dva diska

Podaci se upisuju magnetizacijom čestica u magnetičkom mediju koje jednom namagnetizirane zadržavaju uzajamnim djelovanjem zadani položaj. Na slici se jedan magnetić samovoljno zakrenuo, ali će ga sila odbijanja istoimenih polova natjerati da se okrene na pravu stranu

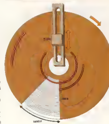
Na stranama 24 i 25 raspravljali smo o memoriji ugrađenoj u kompjutor i povezanoj direktno s procesorom. Pristup memoriji je jednostavan i brz. No kako je ova vrst još uvijek dosta skupa (od 150 do 230 $ za 64K, računajući u cijenu i ostale čipove za kontrolu buseva i osvježenje memorije), trebalo je naći neko jeftinije rješenje — memoriju koja ne treba biti posebno brza. Pojam brzine pristupa podacima (access) mogli bismo ilustrirati s ovo nekoliko papira što su razasuti po vašem radnom stolu u kojima začas možemo naći neki podatak i sa onom gomilom ispisanih fascikala pohranjenih po policama, za koje nam treba i nekoliko sati da bismo u njima nešto pronašli.

Da bi se napravila memorija, potrebna je neka fizikalna pojava koja se može električki potaknuti (koju će kompjutor zabilježiti) i nešto što može izazvati električku pojavu (koju će kompjutor pročitati). U osnovi nije važno kakva je to pojava, pa su se godinama za to upotrebljavale najčudnije stvari. Jedna od najranijih kompjutorskih memorija (sagrađena za kompjutor Mark I, koncem četrdesetih godina u engleskom gradu Manchesteru) služila se serijama električkih pražnjenja zabilježenih na ekran katodne cijevi. Jednostavna, pregledna ali i najpolaganija metoda je bušenje papirnih traka. Mogu se bušiti električnim bušačem, a čitati s pomoću svjetla ili malih električnih kontakata. Mnoge velike kompjutorske instalacije još se uvijek koriste takvim trakama. I zaista, ja još danas čuvam pod svojim krevetom cijeli engleski rječnik dugačak oko 8 kilometara.

Od početka je svakoma bilo jasno da će izumitelj jeftine kompjutorske memorije steći veće bogatstvo nego što je mogao i sanjati. Stoga nije nedostajalo pobuda za istraživanje novih tehnologija, ali jedina metoda koja je izdržala kušnju vremena, i koja je danas općenito prihvaćena, je magnetska. Ako se neka pogodna podloga presvuče magnetičkim nanosom, čestice željeznog oksida što su u njemu mogu se magnetizirati ili ostaviti nemagnetizirane. Na taj način može se zabilježiti binarno 1 ili binarna 0. (U praksi je to nešto kompliciranije. Binarna jedinica bilježi se promjenom smjera magnetizacije sa sjevernoga na južni pol, a za nulu smjer magnetizacije ostaje jug.) Najveća je prednost magnetskog snimanja što svaka čestica magnetiziranog materijala trajno zadržava svoj smjer magnetizacije. Na slici je to prikazano kao polje sastavljeno od malih magneta koji su svi poput vojnika okrenuti u istom smjeru. Ako se jedan od njih pokuša samovoljno okrenuti, ostali mu to neće dopustiti.

Svakom pristojnom magnetu sjeverni pol je uvijek uz južni pol susjednog. Greška može nastati samo ako se jedan magnet nekako okrene pa se njegov sjeverni pol nađe uz sjeverne polove susjednih, a južni, naravno, uz južne. Međutim, svaki onaj koji se ikada igrao s magnetima zna da to uzrokuje međusobnu silu odbijanja i da će nedisciplinirani magnet biti prisiljen da se okrene u istom smjeru i stane u red s ostalima. To je svojstvo koja jamči da će podaci zabilježeni u magnetski medij ostati dulje nepromijenjeni. Vrijeme dakako nije neograničeno. Općenito se smatra da magnetske zapise treba presnimiti svake dvije godine, jer će ih inače pogoditi zlokobna bolest poznata kao 'trulenje bajtova'.





Gore desno: obične audio kazete je najjeftinije sredstvo za pohranu podataka ali je pristup do njih spor i neaspretan. Od tri fajla (file — niz podataka) prikazanih na slici fajl A je na samom početku, fajl B samo što nije očitan, a fajl C se nalazi daleko na kraju kazete

Dolje: disk s podacima usnimljenim na sektore u obliku kriške torte. Glava za snimanje i čitanje može se postaviti na bilo koju od kružnih staza svakog pojedinog sektora. Na taj način može se vrlo brzo doći do podataka

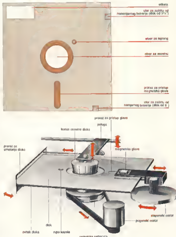
Postoji vrlo malo fizikalnih postupaka s tako dobrim svojstvima, i još uvijek se magnetska metoda smatra najpouzdanijom. Na horizontu se doduše naziru neke nove zamisli (vidi str. 42-43), ali se do sada ništa nije pojavilo što bi uspješno zamijenilo magnetizam. (Ta je tehnika već tako duboko usađena u svijest ljudi da su na primjer strojevi s kolutima magnetske vrpce, iako danas već prilično zastarjeli, postali vizualnim simbolima kompjutora na filmu i televiziji.) Ali upravo serijska priroda, to svojstvo što magnetsku vrpcu čini tako pogodnom za snimanje i reprodukciju glazbe, smeta pri pohrani kompjutorskih podataka. Radi se dakako o stalnom premotavanju i pretraživanju vrpce, jer rijetko treba čitati sve podatke redom, baš onako kako su bili zabilježeni.

Na susjednoj stranici, na slici koja prikazuje audiokazetu sa snimljenim programima, neka kompjutorska igra (A) nalazi se gotovo pri kraju vrpce, program za kućnu blagajnu (B) na sredini, a onaj za konverziju Celsiusovih stupnjeva u Farenhajtove daleko, daleko na početku. Nije zgodno neprestano premotavati i pretraživati vrpce pa su se neki promućurni momci dosjetili da najbolja svojstva magnetske vrpce spoje s prednostima gramofonske ploče. Zato su jednu okruglu ploču presvukli magnetskim nanosom, stavili je na osovinu da se može okretati, a magnetska glava, poput zvučnice na gramofonskoj ploči, pomiče se od ruba prema središtu i obratno. Takvo rješenje omogućuje da glava za čitanje/pisanje u tili čas dođe na bilo koje mjesto ploče, a vrijeme za traženje programa reducirano je na minimum. Iz ove zamisli nastala je jedna dosta komplicirana, a naravno i skupa mehanička naprava.

Disk je podijeljen na sektore (segmente kružnice) i staze (kružnice različita promjera) koji nemaju kontinuirani tok kao na gramofonskoj ploči. Kroz jednu rupicu na disku blizu središta prolazi pri svakom okretaju zraka svjetla dajući signale elektronici koja tako saznaje gdje se u kojem času nalazi glava u odnosu prema sektorima. Sektori se namještaju elektronski, svjetlosni signal se naziva 'timing signal', a cijeli sistem 'soft sector' (mekani sektor). Diskovi se za neke uređaje izrađuju s jednom rupicom za svaki sektor, i taj se sustav zove 'hard sector' (tvrdi sektor). Tako je traženje fajla postalo brzo i jednostavno. Treba samo namjestiti magnetsku glavu na neku određenu stazu i pričekati da se pod nju zakrene pravi sektor. Prosječno vrijeme koje treba glavi da s jedne staze dospije na bilo koju drugu naziva se 'seek time' (vrijeme traženja), a 'latency' (latentnost, skrivenost) je pojam koji označava prosječno vrijeme potrebno bilo kojem sektoru da se zakrene pod glavu.

Flopi disku sa dobro napisanim sistemskim softverom treba prosječno ⅓ do ⅕ sekunde da pronađe bilo koje mjesto na disku, dok je hard disk barem dvostruko brži.

Danas je praktički svaki kompjutor srednje



klase opremljen disk drajvom. Fleksibilni, od plastike napravljeni diskovi, presvučeni su magnetičnim materijalom i uloženi u kvadratične koverte, a izrađuju se u četiri veličine: od 8, 5 ¼, 3 i 3 ¼ inča.

Zbog vrlo precizne mehanike koja minuciozno vodi magnetsku glavu sa staze na stazu postao je ovakav disk drajv skupom igračkom. Pokušalo se s raznim napravama spojiti jeftinoću kazetofona s brzinom pristupa koju ima disk drajv. Komercijalno ime jedne takve naprave je 'The Stringy Flopoy', ali autor nije primijetio da bi svijet bio impresioniran tom napravom. Za pisanja ove knjige, u Britaniji se Clive Sinclair spremao izbaciti na tržište stvar pod nazivom 'Micro drive'. Tvrdi se da radi poput diska, ali s mehanizmom koji vodi glavu spiralnim putem prema središtu i natrag. Taj će hardver biti jeftin, ali i vrlo polagan u usporedbi s flopi diskom.

Diskovi mogu biti jednostrani ili dvostrani, ali

Gore: flopi disk (floppy disk). Sâm disk je tanka plastična okrugla plastična ploča presvučena magnetičnim materijalom, umetnuta u kvadratni plastični ovitak. Površina diska ne smije se dirati prstima, niti se disk smije savijati, grijati, niti se smije pisati po njemu. Ne smije ga se, na primjer, ostavljati ispod telefona, jer ga može uništiti magnetsko polje zvonca. Dozvoljeno je samo, ne vadeći ga nikada iz ovitka, ubaciti ga u prorez disk drajva (disk drive).

Gore: dio mehanizma disk drajva.







Gore od svih mehaničkih naprava do kojih bismo mogli doprijeti najprecizinije je jedinica za pohranu (storiranje) podataka s tvrdim diskom (hard disk). Magnetske glave klize tako blizu površine diska da čak i čestice duhanskoga dima mogu prouzročiti kvar. Zato ova naprava smije raditi samo u potpuno čistom i hermetički zatvorenom kućištu

Većina uređaja s tvrdim diskom sadrži nekoliko diskova na jednoj osovini i sa isto toliko magnetskih glava. Takvo rješenje povećava mogućnost memoriranja ali ne osjetno i cijenu

Desno glava za čitanje i upisivanje podataka na disk je mali elektromagnet. Kada je u funkciji pisanja struja teče kroz namotaj i stvara u prorezu jezgre magnetsko polje koje prodiruć kroz površinu diska magnetizira čestice željeznog oksida. Kada glava ima zadatak da čita magnetizam sadržan u disku inducira u namotaju slabu struju koja se pojačava i vodi u kompjutor

je očito da nema nikakve koristi od dvostranog diska ako uređaj na kojem se obrađuje nema dvostruke glave

Disk drajv može biti jednostruke (single density) dvostruke (duoble density) ili četvorostruke (quad density) gustoće. Taj se pojam odnosi na broj staza koje se mogu upisati ili pročitati. Četvorostruki ili quad density disk bio bi doduše najbolji jer se na njemu može zapisati četiri puta više podataka, ali je s druge strane to najlošije rješenje jer zahtijeva četiri puta veću preciznost pri vođenju glave. Poznato je da quad density disk obrađen na jednom stroju ne funkcionira na drugom iako jednakom. Za prijenos podataka s jednog disk drajva na drugi najbolji su oni jednostruke gustoće. Ali ako ste upućeni samo na vlastiti stroj onda se mirno možete služiti quad density diskom jer će moguća greška kod pisanja i čitanja biti uvijek jednaka

Disk drajv i kompjutor spojeni su višežilnim kabelom. Čitanje i pisanje obavlja se preko DMA čipa (Direct Memory Access — izravni pristup memoriji). To je posebni procesor čija je jedina zadaća da paralelno s glavnim procesorom kontrolira vanjsku memoriju. Brzina prijenosa podataka obično iznosi oko 250 kilobajta na sekundu

Premda hard disk sve više ulazi u upotrebu a i cijene mu naglo padaju još uvijek klasični disk ima stanovitih prednosti. Stari flopi disk je relativno vrlo jeftin može primiti zaista mnoštvo podataka — do 500 KB na svaku stranu može se lako kopirati i bez brige spremiti makar ispod vlastita kreveta a može se poput pisma slati i poštom. Ali je s druge strane vrlo osjetljiv i brzo se troši pa je za volju vlastita mira najbolje uvijek napraviti rezervnu kopiju važnijih fajla

Nezgoda je s mini flopijem što disk zapisan na jednom primjerku, po svoj prilici neće raditi na drugom jer svaki proizvođač pokušava naći neki svoj poseban način zapisa. Premda su svi toga svjesni ipak i novi format od 3 1/4 po svemu sudeći pati od iste bolesti

Na formatu od 8' postoji standard pod nazivom SSSD (Single Sided Single Density disk — jednostrani disk jednostruke gustoće) koji radi prilično dobro na svakom stroju pa se zato komercijalni softver obično distribuira upravo na ovom formatu. Treba dosta truda ako se želi neki softver kopirati s jednog 8' stroja na drugi recimo od 5 1/4'. Najprije ih treba nekako fizički spojiti što obično nije sasvim jednostavno pa onda podatke pretakati preko interfejsa RS 232 (vidi str. 28-29)

U posljednje vrijeme stižu neke povoljne vijesti od proizvođača hard diska. Radi se na tome da više podataka stane na zadani prostor što je manja površina magnetskog zapisa. To znači veći kapacitet diska uz isto vrijeme pristupa jer se glave uvijek pomiču za istu udaljenost

To se može postići samo ako se magnetske glave stave bliže površini diska, što znači da se

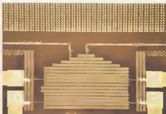
slan fleksibilni disk mora zamijeniti novim vrlo tvrdim i savršeno glatkim. Razmak između magnetske glave i diska iznosi samo 18 milijuntih dijelova inča i doslovce plovi na zračnom jastuku što nastaje zbog okretanja diska. Nečistoća koje uvijek ima u zraku, prašina, pa čak i čestice duhanskog dima, moraju se sasvim isključiti da bi uređaj radio ispravno. Stoga je mehanizam stavljen u hermetički zatvoreno kućište s kondicioniranom atmosferom. To je pravo čudo tehnike u koje se može pohraniti preko 50 megabajta podataka, a nije ništa veće od standardnog kućišta 5 1/4 flopi diska. Vrhunska tipkačica može napisati oko 1000 riječi na sat, što znači da bi se na prostor od 30 MB mogao smjestiti rad od 623 radna dana — više od dvije godine rada.

Pomama za memorijama sve većeg kapaciteta ne zamire. Jedna od zastrašujućih shema predviđa da se sa horizontalnoga zapisa u kojega se područja magnetizacije nalaze na površini diska, pređe na vertikalni, gdje bi područja magnetizacije tekla s jedne strane diska na drugu. Smatra se da bi taj način povećao gustoću zapisa oko 40 puta. Jedan mini flopi sadržavao bi tada oko 20 MB podataka, a hard disk 1200 MB — osamdeset godina tipkanja. Slovom i brojem!

Magnetski mjehurići

Danas već mnogi smatraju da bi se moralo naći neko elegantnije i kompaktnije rješenje nego što su komplicirane mehanike i diskovi okretani s motorima. Nije najveća smetnja u samom disku već u mnoštvu pokretnih mehaničkih dijelova sklonih kvarovima. Osim toga, magnetska se glava često prlja, što ima katastrofalne posljedice za zapisane podatke. Rješenje o kojem se mnogo govorilo kasnih sedamdesetih godina je memorija s magnetskim mjehurićima (magnetic bubbles).

Suština ideje je u tome što neke vrsti magnetskih materijala omogućuju formiranje i pomicanje malih područja koja se razlikuju od svoje okoline u smislu postojanja magnetskog polja. Takva područja nazivaju se magnetskim mjehurićima ili domenama magnetizma. Postojanje mjehurića označava logičku jedinicu, a nepostojanje logičku nulu. Mjehurići se stvaraju na jednom kraju magnetskog materijala, a izlaze iz njega na drugom kada se na njih djeluje magnetskim poljem. Umjesto da se mjehurići fizički kreću ispred magnetske glave, oni putuju duž magnetskog materijala, pa tako dolazimo do memorije bez ikakvih pokretnih dijelova. No bilo bi dakako prelijepo da su stvari tako jednostavne. Da bi se realizirala ovakva memorija, treba izraditi vrlo fine zavojnice koje troše mnogo energije i zasada su vrlo skupe. Konvencionalni diskovi proizvode se u ogromnim količinama i zbog toga su jeftini, pa je pitanje da li će im memorije s magnetskim mjehurićima ikada moći konkurirati. Zasada se ta nova vrsta memorije najviše upotrebljava u kompjutorima na ratnim avionima i brodovima, gdje bi diskovima smetala prašina i artiljerijska vatra — a kad je u pitanju vojska, ne pita se za cijenu.



Tehnologija pohrane podataka s pomoću magnetskih mjehurića (magnetic bubbles). Umjesto da se podaci upisuju u magnetički medij rotirajućeg diska, uređaj s magnetskim mjehurićima provodi podatke kroz nepomični magnetski medij. Redovi horizontalnih paralelnih pruga su staze duž kojih se magnetizirana područja pomiču djelovanjem snažnog vanjskog namotaja. Vertikalne prečke predstavljaju neku vrst glave za čitanje i pisanja.

Dolje: jako povećana glava na površini tvrdog diska. Magnetička tvar (na dnu stranice) je smeđa, glava je bijela (ostatak stranice), a razmak između njih širok 18 mikrona je plav. To je tako mali razmak da presjek ljudske kose prikazan u istom omjeru (zasjenjeno područje) zauzima veću površinu od cijele stranice ove knjige.

FAJL I OPERATIVNI SUSTAV

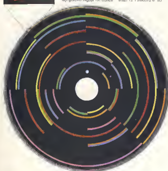
Disk je nešto poput praznog uredskog ormarića koji bi bio besmisleni predmet kad ne bi imao ladice s natpisima da se što brže može naći mjesto za svaku odloženu stvar. U kompjutoru ovu funkciju obavlja tzv. 'operativni sustav' (dalje ćemo upotrebljavati kraticu OS), program čiji je zadatak da vodi brigu o cijelom kompjutorovu kućanstvu.

Na najnižoj razini OS vodi mnoge naoko sitne, ali ipak vrlo važne poslove. Pogledajmo što će se dogoditi ako napišemo ovaj mali program:

10 INPUT Unesi sljedeći broj, N

Programski jezik BASIC kojim je ovaj program napisan također je program, ali onaj koji omogućuje komuniciranje između korisnika i kompjutora. Ako otipkamo RUN, naredbu koja određuje izvršenje programa, operativni sustav će je posredstvom BASICa primiti, ispisati na ekranu rečenicu "Unesi sljedeći broj", pričekati da se preko tastature unese broj i pritisne tipka RETURN. Nakon toga OS opet izručuja upisani broj BASICu.





Shematski prikaz diska s podacima upisanim u koncentrične staze raslevljene na sektore. Svaki fajl podataka prikazan je u drugoj boji. Što se više podataka upisuje i briše, slobodni prostori na disku sve su rascjepkaniji, pa se isti fajl gotovo nigdje ne doliče svoga nastavka na drugom sektoru. Na upravljačkoj stazi (posebno povećani kvadrat) ubilježeno je mjesto na kojem se nalazi svaki pojedini fajl. Na primjer, fajl pod nazivom CALK nalazi se na stazi 4, sektor 7, a nastavak na stazi 12 i sektoru 6 itd.

Operacija ispisivanja rečenice "Unesi sljedeći broj" mogla bi također biti kodirana u BASICu. Ali da bi se na istom računalu mogli obrađivati programi pisani na ostalim programskim jezicima ili direktno u strojnom kodu, nije tako učinjeno. Ovako se može standardizirati način na koji informacija dolazi ili se vraća od operativnog sustava, što garantira nesmetanu obradu standardnih softverskih paketa.

OS upravlja i printerom: šalje mu tekst ispravnom brzinom i prepoznaje signale rukovanja (handshaking — vidi str. 28-29).

On će također obaviti sav onaj složeni posao oko zapisivanja informacija na disk ili čitanja s njega. Postavlja se pitanje najboljeg korištenja raspoloživa prostora: jer dok je disk prazan, to je jednostavno, najprije se upisuje prvi fajl podataka, zatim drugi, treći itd. sve do središta kada je cijeli ispunjen. Sigurno ste se već ranije zbog nekog razloga morali vratiti prvom fajlu. Ispraviti nešto, izbrisati staru verziju i zapisati novu. Vjerojatno nova verzija neće sasvim pristajati u stari prostor: bit će veća ili manja, pa će na disku vrlo brzo nastati prava zbrka popunjenih i praznih prostora, premalenih da bi se u njih moglo upisati nove fajlove, a prevelikih da bi ostali prazni.

Da se prostor diska može u potpunosti iskoristiti, fajlovi se cjepkaju na odsječke, pa je uvedena, obično oko vanjskog ruba diska, posebna upravljačka staza (directory track). Na nju se upisuje položaj i odsječak svakog fajla. Operativni sustav tako saznaje da je, na primjer, fajl 1 zapisan na odsječcima 34, 35, 36, 47, 53, 96, 97 i 98. Fajl 2 može biti zapisan na odsječcima 2, 3, 4, 5, 6, 7, 26, 29, 39 i 126. Fajl 3 je na drugima, dok su svi ostali slobodni. Kada se jedan fajl izbriše, odsječci što ih je zauzimao označeni su kao prazni. Na taj način, uz uvjet da postoja upravljačke staze i skakanje magnetske glave s jednog odsječka na drugi, može se površina diska maksimalno iskoristiti.

Korisnik zapravo ne treba o tome mnogo znati. On samo traži da operativni sustav čita i zapisuje podatke što on i čini, a kako to čini, njegov je problem.

Prije no što krenemo dalje, bilo bi dobro pogledati što predstavlja pojam fajl (file). To je jednostavno dugački niz bitova (što bi drugo mogao i biti) zapisan na disku sa svojim imenom, svojim početkom i svršetkom. Naziv se upisuje na upravljačku stazu uz broj odsječka na kojem se fajl nalazi. A da bi se točno znalo gdje fajl završava, na kraju se upisuje EOF oznaka (End Of File — svršetak fajla). Za vrijeme čitanja svakog fajla, OS stalno ispituje da li se u nekom bajtu nalazi EOF oznaka, pa kad na nju naiđe, čitanje se prekida. Ako se kojim slučajem EOF oznaka nađe u sredini fajla, nastaju poteškoće.

Bajtovi u jednom fajlu mogu se promatrati na dva načina: kao podaci ili kao program. Podaci mogu predstavljati neki tekst. Ili brojeve — re-

zultate nekog proračuna. Tekst i brojevi mogu biti izmiješani, a mogu predstavljati i koordinate točaka neke kompjutorske grafike. Teško je na prvi pogled odrediti što neki fajl predstavlja, jer dobiva smisao tek kada bude pročitan uz pomoć pravog programa.

Ako se radi o nekom programu, bajtovi jednog fajla interpretiraju se kao naredbe strojnog koda (vidi str. 80-85) i kao memorijske adrese. Naći će se tu svakako i nešto podataka razasutih po programu, ali programski bitovi će uvijek razlikovati podatke od programskih naredbi. Za interpretaciju se brine OS, a vodi ga ime fajla. Po tradiciji naziv fajla sastoji se od dva dijela: od naziva kojim se sâm fajl identificira i dodatka koji saopćava operativnom sustavu vrstu fajla.

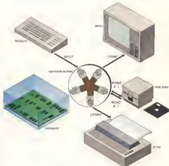
Kada se operativnom sustavu da naziv nekog programskog fajla, on odmah zna šta treba s njim uraditi. Ako ga treba samo pročitati sa diska, zna da ga treba unijeti u memoriju i postaviti početak tamo gdje mu je mjesto. Za sve ostale vrste fajla treba mu neki program, jer inače ne zna što da s njima započne.

Dobar OS može još mnogo više od toga. On sadrži mnoge male izv. epizodne (utility) programe. To su jednostavni programi koji, na primjer, javljaju koliko je još slobodne prostore ostalo na disku ili koliko je veliki fajl. Mijenjaju imena fajlovima i kopiraju ih s jednoga diska na drugi. Omogućavaju korisniku da se služi (i to ispravnom brzinom) printerom. Ako je sustav predviđen za više korisnika (vidi str. 146-149), sprečava da se fajlovi jednoga miješaju s fajlovima drugog. Dijeleći fajlove korisnicima, budno pazi da ikogod ne pokuša čitati ono što nije njemu namijenjeno. Nadalje, omogućava izmjenu informacije između korisnika i njihovo komuniciranje s vanjskim svijetom.

U kompjutorskom svijetu postoji nekoliko grupa operativnih sustava čiji su proizvođači nasljedili shemu velikih kompjutorskih kompanija, po kojoj svaki proizvođač automatski snabdijeva svoje kupce vlastitim operativnim sustavom. Najvažnije im je spriječiti korisnike da slobodno kupuju softver tamo gdje je najjeftinije. Treba znati da je softver roba koja danas donosi najveće profite, veće nego one od hardvera.

Iako proizvođači ugrađuju u svako individualno računalo OS koji najbolje služi svojoj svrsi, ipak OS koji je vlasništvo samo jednog proizvođača ne može potpuno ispunjavati prednosti osnovne ideje da zajednički OS čini različita računala kompatibilnima.

Otkriće OSa nastalo je gotovo slučajno sredinom sedamdesetih godina, kada je Gary Kildall razvijao softver za zapis i čitanje sa diska za jedan mikrokompjutor. Nazvao ga je CP/M (Control Printer/Monitor). Uspjeh je bio senzacionalan, pa su ga prihvatile desetine, a kasnije i stotine proizvođača 8080 i Z80 mikrokompjutora. Tokom vremena ovaj je akronim promijenio osnovu, danas CP/M znači Control Program for



Microcomputers' (kontrolni program za mikrokompjutore).

Zajednički OS, kao što je to CP/M, osigurava tržištu softvera mnogo širi krug potrošača. Sistem specifičan samo za jedno određeno računalo, pa čak i mnogo snažniji od CP/M, značio bi isto što i tiskati knjigu na finskom umjesto na našem jeziku, jer je sadržaj primjereniji logici finskog jezika. Ali, na žalost, ima premalo čitalaca koji čitaju finski. Tako i razvoj masovnog tržišta softvera ne bi bio moguć bez zajedničkog operativnog sustava.

Na žalost, vlasništvo jednog široko rasprostranjenog operativnog sustava tako je unosno da ga se svim silama pokušava ozakoniti. Posljedica je rascjepkanost tržišta. Nedavno je jedan proizvođač odlučio promijeniti svoj OS kako bi omeo proizvođače softvera izvan vlastite tvrtke. Reakcija korisnika je bila vrlo slaba, a sam je proizvođač prošao još lošije.

U ovom času na tržištu postoje samo dva (ili bolje reći jedan pravi i dva slabašna) sustava. Glavni takmac je CP/M i njegovi poboljšani nasljednici (kao MS-DOS). CP/M je spretan i jednostavan, doduše prema standardima velikih kompjutorskih sustava čini se prilično slabašan, ali za individualnog korisnika je odličan. Postoji nekoliko sličnih sustava, kao na primjer TurboDos, MacNos i MMos (vidi str. 148-149).

Glavni rival CP/M i MS-DOS sustavu je UNIX sa svojim derivatima. U početku, prije nekih desetak godina, bio je zamišljen kao sustav za visokonsnečke minikompjutore u Bellovim labora-

Posebni program koji nazivamo operativnim sustavom (vrti se unutar kompjutora, iako je ovdje shematski prikazan izvan njega) povezuje procesor, tastaturu, ekran, disk drajv i printer u jednu cjelinu. Sva računala istog operativnog sustava služe se identičnim programima. Natpisi velikim slovima predstavljaju naredbe iz programskog jezika Microsoft BASIC, a predviđene su za standardne operativne sustave i mogu nepromijenjene funkcionirati na više različitih kompjutora.

tonjima Ako je CP/M preskroman, onda je UNIX pretjerano raskošan U svom sadašnjem obliku nemijenjen je profesionalnom programeru omogucujuci mu da pritiskom jedne tipke stavi u pogon sve postupke nekog programa Izlaz jednog programa može se pretakati na ulaz drugoga Opramljen je sjajnim mehanizmom nazvanim Shell (školjka) koji omogucuje da jedan program stavlja u pogon drugoga kao da se radi o operativnom sustavu

UNIX služi profesionalcu veoma dobro ali u svom cjelovitom obliku zbunjuje individualnog korisnika Čini se da sada na tržišlu dolazi do ukrštanja UNIXa i CP/Ma Svakako treba nešto učiniti da se UNIX ponaša poput CP/Ma kad korisnik to hoce pa da se softver za CP/M može tjerati I pod UNIXom

Smelltalk

Do sada smo promatrali operativni sustav na njegovoj najnižoj razini kad poput nadzornika nad kompjutorskim kucanstvom pazi na disk tastaturu ekran i pnnter Međutim OS ima i odgovornost na jednoj višoj razini jer je on jedino sredstvo koje upravlja programima

Moguce rješenje je da se sastavi OS koji ce automatski natočiti cijele serije programa do dati im još i naredbe koje bi korisnik inače morao pronalaziti u priručniku i unositi ih preko tastature U Xeroxovu istraživačkom centru u Palo Altu razvijen je novi sustav a preuzeo ga je Apple za svoje novo racunalo Lisa sustav koji odbacuje alfabetski prikaz programa i podalaka i nudi korisniku jednostavnije rješenje To je SMALLTALK koji korisniku predocuje slike postupaka umjesto njihovih verbalnih I numeričkih opisa Na pnmjer, slikom fascikle predstavljen je fajl podetake slika printera znači printer a kante za olpatke — stvari kojih se treba rješiti

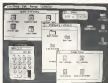
Da se pomoću tih slika obavi neki koristan posao kursor se vodi po ekranu s napravom nazvanom mouse' (miš vidi str 28-29) Dok pomičete miša po slolu kursor se u istim smjerovima pomiče po ekranu i kada se nađe na slici simbola radnje koju želite obaviti pritisnete dugme na mišu i — to je sve Ako želite da printer otipka fajl podataka treba samo kursor dovesti na spisek fajla pritisnuti dugme na mišu zatim kursor na sliku printera pa opet pritisnuti dugme i naredba je izvršena

Dakako za takvo rjesenje treba imati procesor velikih mogucnosti a i ekran mora biti visoke rezolucije da bi se na njemu mogle prikazati slike sa svim potrebnim detaljima Lisa je opremljena s vrlo bogatom dokumentacijom programa koja se može listati na ekranu poput listova papira Može se premjestiti samo dio jedne stranice na drugu ili jedna ispunjene stranice nalijepiti' na neku sliku (vidi str 100) Postoji softverski paket za crtanje onaj za obradu teksta (word processor) a dakako i mnogo veći programi za obradu podataka (data base manager)

Sve to postavlja velike zahtjeve na hardver Lisa radi s najsnažnijim 16-bitnim procesorom 68 000 i 2MB RAMa pa je zbog toga ovo računalo vrlo skupo Sudeći po reakcijama pri demonstracijama Lise prosječni korisnik je još uvijek vrlo naivan Treba pričekati da se vidi kako će dugo ljudi radije biti spremni platiti veliku cijenu za hardver kojim se imitiraju papiri na ekranu nego da provedu nekoliko sati nad knjigom učeci kako da to izvedu na ekonomičniji način

Ja međutim najozbiljnije mislim da se takve stvari mogu okrenuti u vlastitu suprotnost Premda dopadljive ovakve metafore kompjutorskih pojmova mogu potaknuti neiskusna korisnika na kupovinu ali na duže staze to može izazvati suprotni učinak Kao što pokusavam dokazati u ovoj knjizi a pokušavat cu i dalje da postoje mnogi vidovi kompjutorstva koji ne trpe usporedba sa svakidasnjim shvacanjem pojma informacije na papiru Prije ili kasnije oni koji zaista žele biti kompjutorski obrazovani, morat ce shvatiti da rad na kompjutoru ne može biti isto sto i rad s olovkom i papirom

Ljudi navikli na olovku i papir teško shvaćaju što je zapravo kompjutor. Objasniti im to jednako je teško kao kočijašu volovske zaprege objasniti probleme auto cesta, prometnih petlji i signalizacije, parkiranja, goriva, mjenjačkih kutija, guma, svjećica i još mnogo čega drugog. Nakon prve vožnje automobilom on će najvjerojatnije napraviti saobraćajni prekršaj i izgubiti zanimanje za motore. Apple pokušava riješiti taj problem lažno prikazujući svoje računalo kao volovsku zapregu. Lisa je pokušaj pretvaranja ekrana u površinu radnog stola.



Evo kako izgleda Lisin ekran: na njemu se nudi niz predmeta kućne upotrebe: prazan papir, kalkulator, košara za otpatke itd. Prema autorovu mišljenju takva rješenja mogu voditi samo u konfuziju.



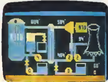
Uobičajena slika tokom osamdesetih godina bračni par kontrolira kućnu blagajnu na kompjutoru i nikako ne može shvatiti kuda je otišao novac

Eksplozivni razvoj proizvodnje jeftinih čipova izazvao je iznenađujući prodor malih, kućnih ili osobnih' kompjutora na tržište. Dok ovo pišem (sredina 1983), ta klasa računala ima prilično jednostavni hardver. Procesori su ZX80 ili 6502, na ekranima prikazuju 40 × 20 znakova, a kadšto i manje, a neki imaju samo velika slova i seriju zgodnih, ali primitivnih grafičkih znakova. Gotovo svi imaju sliku u boji, što je zgodno ali pod uvjetom da vlasnik ima televizor u boji i da nitko od ukućana te večeri ne želi gledati popularnu seriju na tekućem programu. Često imaju manje od 64K RAMa, a najčešće između 16 i 48K. Mana im je što uglavnom nemaju diska. Podatke spremaju na obični kazetofon ako je ispravan, a ako nije ... u svakom slučaju pristup podacima je jedino serijski (vidi str. 22-23). Dodajmo još da je mikrokompjutor zaista upotrebljiva naprava samo ako ima više od 20K slobodne memorije.

Zbog tih ograničenja korisnik kućnog kompjutora prisiljen je da radi samo s malim programima koji se isključivo vrte u unutarnjoj memoriji. Posljedica toga je ogromno tržište softvera za mala računala, izraslo gotovo preko noći. Najviše je tu igara. Šah je među njima najintelektualniji, Molar Mauler najbeznačajniji. Nakon igara postaju sve popularniji edukativni programi koji pokušavaju posredstvom igre objasniti osnove algebre, fizike ili nekog (govornog) jezika.

Neko vrijeme se mislilo da će se mala računala uglavnom upotrebljavati u školskoj obuci te da će kompjutorizirani školski satovi uspješno nadmašiti udžbenike. Istina je da kompjutor može jače vezati učenikovu pažnju, može ga ispitivati i ocjenjivati napredak i sposobnosti. Ali je nevolja u tome što đaci u prosjeku nisu imućni ni osobno, a niti je društvo spremno da im takvu opremu stavi na raspolaganje. Zbog toga im škole mogu pružiti samo jeftina računala, a ta su opterećena svim, malo prije opisanim ograničenjima. Posljedica toga je vrlo skromni softver koji nije u stanju nadomjestiti ni najjednostavniji udžbenik.

Još je ozbiljnija primjedba da se učenje ne





sastoji samo u prolaženju školskoga gradiva niti u testiranju učenikova stupnja razumijevanja predmeta. Dobar nastavnik ima mnogo prisniji odnos sa svojim đacima i on točno zna što oni misle da znaju, što ne znaju i kakve probleme imaju pri učenju, pa prema tome prilagođava ritam nastave. On uči o svojim učenicima isto onoliko koliko učenici uče o predmetu koji im predaje. Za sad još nema te umjetne inteligencije, ili barem nema onoga koji bi znao napisati tako savršene programe koji bi nadomjestili dobrog nastavnika. Sve su to razlozi da kompjutori igraju tako malu i sporednu ulogu u školskoj nastavi.

Ali zato kompjutorsko obrazovanje kao predmet ulazi sve sigurnije u škole, a jeftina računala su pogodna upravo za tu svrhu. Poznato je da ima nastavnika koji se tome strastveno opiru. Pretpostavljam da je razlog u tome što djeci ne treba mnogo formalne poduke o komp-



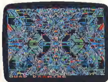
jutoru Brzina kojom računalo reagira na njihova pitanja podsiče ih da pokušavaju sve više i više pa se uloga naslavnika svodi na povremenog savjetnika a neki to smatraju poniženjem

Gotovo sva mala računala opremljena su nekom verzijom BASICa da bi korisnik zamoren gotovim igrama počeo kreirati vlastite Treća kategorija softvera je jedna od verzija programskih jezika kao LISP i FORTH Konačno pokušavalo se malim računalima prilagoditi i ozbiljnije poslovne softverske pakete (vidi str 94-101)

Usprkos nedostatku diska serijskom zapisivanju podataka na magnetsku vrpcu i svim ostalim hardverskim ograničenjima tržište softvera za mala računala nastavlja bujati pa je iako samo tri godine nakon pojave prvog zaista jeftinog računala — Sinclairove ZX80 — samo u Velikoj Britaniji bilo preko milijun kućnih kompjutora Sredinom 1983 od svakih 5 domaćinstava jedno je imalo mikrokompjutor a tržište softvera počelo je ozbiljno ugrožavati primat tržišta pop glazbe i videa Čak se i vodeća glazbena kompanija kao što je Virgin Records bacila na posao s kompjutorskim igrama I tako su se sada odjednom u novinama počele pojavljivati slike golobradih milijunaša koji su doduše propali u školi, ali su zgrnuli prava bogatstva izmišljajući kompjutorske igre

Slika novokomponiranog kompjuteraša nimalo nije nalik onim mučaljivim čudacima što smo ih mogli vidjeti one davne 1970 godine kako se motaju uokolo s punim rukama elektronike i punim glavama strojnih kôdova

Sigurno je da neće proteći mnogo vremena do osobnog mikroračunala koja će biti opremljeno 16-bitnim procesorom i diskom kao vanjskom memorijom Imat će grafiku visoka rezolucije i gomilu RAMa Prodavat će se u velikom broju primjeraka, a cijena će mu biti jedva nešto viša od današnjih malih računala Tržište osobnog softvera još će se jednom napregnuti nudeći ono što danas nazivamo skupim profesionalnim softverom (vidi str 94-101) ali vjerojatno ne za stotinu već za desetak funti ili dolara



KOMPJUTORSKE IGRE

Da li je Babbage Turing ili Von Neumann ikada mogao sanjati da će borba sa Svemirskim osvajačima biti jedno od najraširenijih područja upotrebe kompjutora?



Veći dio čovječanstva uvjeren je da su kompjutori izmišljeni samo za igru što dakako nije točno dok je za ostale i sama pomisao na kompjutorsku igru dovoljan povod za neraspoloženje Naravno da ni ovi drugi nisu u pravu

Kompjutor je uistinu izvanredno sredstvo za neke vrste igara jer se njime mogu vrlo brzo organizirati vrlo komplicirane igre Često se naklapa o sadističkim tendencijama koje nesumnjivo postoje kod nekih autora ali i to je pretjereno jer kompjutorske igre traže minimalno psihičko sudjelovanje iako mogu biti uzbudljive poput nogometnih utakmica

Postoji cijeli spektar kompjutorskih igara od onih nevinih poput Svemirskih osvajača (Space Invaders) koja traži samo dobre reflekse do krajnje intelektualnih kao što je šah Između tih dviju krajnosti nalaze se zanimljive igrarije koje ne traže brze reflekse ali im treba bistra glava U nekim se igrama igrač stavlja u poziciju komandante koji u ograničenom vremenu treba da donosi strategijske i taktičke odluke ali kako to često biva u životu, bez potpunog poznavanja činjenica

Otkada ima kompjutora, ima i kompjutorskih igara Povijest im seže do samih početaka i prve su nastale kao zabranjene zabave programera velikih poslovnih i sveučilišnih sustava sastavljene u stankama između napora dnevnog posla Stare i poštovene Zvjezdane staze (Star track) vjerojatno su predšasnik svih kompjutorskih igara i zacijelo u svijetu nema kompjutora koji taj program nema pohranjen negdje u svojoj memoriji Što više pametne kompjutorske organizacije stimuliraju svoje programere na sastavljanje igara smatrajući da takve aktivnost unatoč troškovima navodi na kreativno eksperimentiranje

Drugi još časniji predšasnik kompjutorskih igara je tzv tehnika simulacije (vidi str 88-89) Njome se najviše služi vojska u ispitivanju novih oružja i strategija Tu je kompjutor često sveden na ulogu knjigovođe koji ogromnom brzinom izračunava rezultate mogućih strategijskih odluka

Najveća opasnost krije se u činjenici da su ljudi spremni vjerovati u simulaciju iako negdje u samoj logici stvari mora postojati neko slabo mjesto što cijelu stvar čini opasnom

Najbolji primjer za tu tvrdnju je kompjutorska simulacija svjetske privrede koju je 1970 godine sastavio Rimski klub (grupa poslovnih ljudi političara i civilnih službenika) Oni su izdali onaj famozni Izvještaj pod nazivom Ograničenje rasta koji predviđa kolaps svjetske privrede 2000 godine kada će glad i bezvlašće pogoditi jednako i siromašne i bogate Kako je to strašno proročanstvo poteklo od vrlo ugledne grupe a i svijest da je pri tome upotrebljen kompjutor učinilo je ovaj izvještaj vjerodostojnim

Kasnije se pokazalo da je napravljene fatalna greška pri kompjutorskoj simulaciji Naime u jednoj važnoj sekciji koja je sadržavala cijene i rezerve sirove nafte u svijetu (kao što znamo rezerve sirove nafte u zemljinoj utrobi su ograničene i potrebno je što prije tražiti nove izvore energije) u programu je izostavljena simulacija razvoja cijena u slučaju da zalihe presuše Programom su bile definirane cijene kakve su bile početkom 1970 i nisu mijenjane sve do dana kada će nafta sasvim presušiti Naravno da je takva pretpostavka morala dati sliku opće pometnje u svjetskoj privredi

U stvari kako rezerve nafte presušuju proizvođači podižu cijene da bi kompenzirali smanjenje dohotka u budućnosti a ovo opet automatski stimulira razvoj alternativnih izvora energije Ako imalo vjerujemo u moć tržišta onda je sigurno da će svijet preći na bilo koju drugu energiju i to s vrlo malim zakašnjenjem i izbjeći potpunu katastrofu kao što je to bio prorekao kompjutorski simulator

Mikrokompjutorske igre se mogu klasificirati u tri grupe u nervozne romantične i intelektualne One nervozne traže brzo oko i sigurnu ruku ali malo pameti Prati ih jedinstveno uzbuđenje koje stvara psihološka petlja' što teče preko oka i ruke a zatvara se izvan igrača u stroju

Romantične igre vuku svoje podrijetlo iz literarnog žanra nastalog po uzoru na pripovijetku



Gospodar prstanova (The Lord of the Rings) J R R Tolkiena Iz te se sheme razvila cijela grupa pustolovnih igara Jezgra je uvijek nekakva neobična scenografija (može biti i trodimenzionalna) podijeljana u ćelija Svaki igrač postaje jedno lice — Čarobnjak Ratnik Warlock Troll — i bira predmete koje ce trebati za svoju pustolovinu svjetiljku čarobni mač ključ ili bocu s vodom U svakoj fazi igre može se ići gore ili dolje Istočno, zapadno sjeverno ili južno u drugu ćeliju Prolaz u nju može biti zapriječen preuskim vratima kroz koja se ne može iznijeti vreca zlata što je igrač nosi ili ce trebati ključ da bi se ušlo u ćaliju a igrač ga nije ponio

Kad uđe u ćeliju tamo ga čeka ovisno o sreci ljudoždar ili prekrasna djevojka Igra može trajati danima a one najmodernije opremljene su sjajnom koji put i animiranom grafikom

Treća grupa igara ona Intelektualna bavi se tek malo atmosferom a težište je u odgonetanju strategije Tipični je primjer Istočni front gdje igrači preuzimaju uloge generala upravljajuci oklopnim armijama u ratu protiv nacista Akcent





je na dinamici u kratkom vremenu treba pokrenuti divizije tenkova a kada krenu treba ih na vrijeme zaustaviti Pri sastavljanju ovakvih intelektualnih igara velika pomoć autorima je tzv servo petlja' (servo loop — vidi str 130-131)

Hammurabi je također intelektualna igra u kojoj je igracu dano da vlada feudalnom državom Početkom svake godine vladar raspolaže s izvjesnim brojem podanika broj kojih je smanjen za umrle i odbjegle a povecan za novorođene Na zalihi ima žita manjeg za kolicinu što su je pojeli štakori Ima i dvoraca vojnika tržnica i palača Vladar donosi odluke o visini poreza i predviđa razinu korupcije među činovnicima određuje količinu žita koju ce podijeliti seljacima ili sačuvati za sjetvu a može unajmljivati još vojnika i graditi još palača Ali tada ga napada susjedni vladar kojega igra kompjutor Ovisno o snazi svoje vojske igrač može pobijediti i zauzeti dio susjedove države, ali može izgubiti i rat i zemlju Posljedice odluka donesenih jedne godine očitovat ce se tek nekoliko godina kasnije To je vrlo slozena i vrlo uzbudljiva igra

Sve bi te igre bile mnogo privlačnije da se na ekranu može prikazati živi animirani scenski prostor Ali kako ce se vidjeti na str 112-115, animirana grafika postavlja na kompjutor velike zahtjeve te su klasična 8-bitna računala u tome gotovo nemocna Posebnih prednosti pak imaju 16-bitni procesori upravo kada se radi o igrama i simulacijama

Simulacija leta lakog aviona tvrtke Microsoft za osobno računalo IBM sa svojim kokpitom Instrumentima, koji rade potpuno realistički i s pogledom kroz staklo kabine tako je sugestivna da je gotovo jednako dosadne kao i letenje pravim avionom osim osim ako ponestane goriva i avion se razbije







Ima jedne vrlo stara priča što je učitelji rado pričaju ucenicima a govori o davnim vremenima kad je Aleksandar Veliki bio mlad lijep i svemocan gospodar ogromne države U matematici ga je podučavao Aristotel, i jednoga dana umoran od ucenja Aleksandar zapita učitelja da li postoji neki laganiji način učenja geometrije? Ovaj mu je na to mirno odvratio da nema kraljevskoga puta u geometriju Svatko bio bogat ili siromešen mocan ili slab mora učiti na isti nacin

I kompjutor nije iznimka Duboko sam uvjeren da će tokom osamdesetih godina kompjutorska naobrazba biti jednako važna kao što je pismenost postala važna nakon Gutenbergove izuma — nakon pojave tiskarstva ostati napismen bilo je jednake mana kao šepavost

Biti kompjutorski naobražen znači u prvom redu upoznati kompjutorske konvencije naučiti na primjer da tipka RETURN pokrece lanac događaja u računalu Drugo to pretpostavlja razumijevanje mnoštva zbivanja koja se događaju u računalu ali se ne mogu vidjeti kao sto su memorijske lokacije programi ili podaci

Kompjutor može obavljati samo mali broj praktičnih radnji od kojih je vecina ograničena na uski prostor njegova kućišta Vecina onoga što prelazi te granice i seže u vanjski svijat najčešce se svodi na pisanje pisama i konstruiranje slika na ekranu ili papiru

Mjereno ljudskim mjerilima kompjutor je invalid koji doduše može čitati ili crtati ali je potpuno nepokretan pa čak i slijep Zato treba u prvom redu biti realističan u odmjeravanju svojih želja jer se često vjeruje da računalo može pomoci u svakodnevnim poslovima a možda čak pripremiti i — jelovnik Nije to loša zamisao ali je na žalost neizvediva Ako vjerujete da



kompjutor može pogledati u smočnicu i hladnjak i javiti vam da tamo ima još tri riblje konzerve i napola prazna staklenka majoneze te da se od toga jedino može napraviti neka vrst paštete — prevarili ste se. Ako još zaželite zapitati a šta je sa salatom, a računalo neka vam odgovori da je salata neupotrebljiva jer je mačka učinila neku nepodopštinu po njoj, dvostruko ste se prevarili.

Jer upravo to kompjutor ne može. Siromašna beznoga i slijepa naprava ne može gurati nos u smočnicu, jer jednostavno nosa nema. Da bi ga se nahranilo i napojilo podacima, vi morate biti njegovo oko i njegove noge, njušiti po smočnici i hladnjaku, pa još i vlastitim prstima otipkati podatke. Kada sve to obavite sami ste postali perfektan kompjutor za pravljenje jelovnika opremljeni još i osjećajem gladi koju silikoni nikada neće moći osjetiti.

Područja praktične djelatnosti kompjutora ograničeno je, bar za sada, na baratanja simbolima po ekranu i papiru, ali i to je mnogo, jer pokriva prilično široko polje intelektualnih aktivnosti — znanost, privredu i zabavu.

U pisanju programa najvažnije je unositi podatke koji su podjednako razumljivi i vama i kompjutoru. Svemirski osvajači sjajan su primjer za to. Na ekranu se vide sličice malih monstruma koje treba poubijati. Kompjutor ih shvaća kao oblike koje treba razasuti po ekranu, pa kada se koordinate ispaljenih projektila poklope s koordinatama čudovište, računalo odmah zamjenjuje sliku čudovišta sa slikom destrukcije i istodobno se javlja gromkom eksplozijom iz zvučnika.

Kontrolirajući svoju blagajnu, čovjek može nakljukati računalo brojevima koji za njega znače dobitak ili gubitak, napredak ili propast, ali za kompjutor su to samo puki brojevi i ništa više. Gubitka jednostavno iskazuje s predznakom — pa ako njegov vlasnik nakon toga ode na tavan i tamo sebi omota uže oko vrata, kompjutora se to neće nimalo dojmiti.

Programski jezici samo su zbirka simbola što osiguravaju međusobno sporazumijevanje. S jedne strane su vrlo pametni, ali s druge mogu biti i sasvim glupi. Oni mogu logaritmirati, izračunavati sinuse i kosinuse i raditi još mnoge pametne stvari, ali uopće ne mogu organizirati rješenje nekog problema.

Iako ima mnogo toga što kompjutor ne može, ipak ima i mnogo onog što može. Pogledajmo, na primjer, što sve mora učiniti da bi izračunao broj proteklih dana između dva datuma. Zadatak u prvi mah jednostavan, ali zapravo da bi ga se riješilo treba organizirati jedan vrlo pametan program.

Na početku treba naglasiti da Amerikanci ispisuju datume na staroengleski način: najprije mjesec, zatim dan, pa na kraju godinu. Evropljani to rade drugačije: dan, mjesec, godina. Program dakle mora biti takav da odmah na početku može razlučiti o kojem se načinu radi (jer datum 4 3 84 može značiti 3. travnja za Amerikanca, a 4. ožujka za Evropejca). Osim toga mora svladati još jednu moguću zapreku, a ta je da se datumi mogu pisati na nekoliko načina: 3 travanj 1982, 3 4 82 ili 3/4/1982. Ne tiče ga se da li su brojke odvojene razmacima, točkama, crtama ili na bilo koji drugi način. Ako su mjeseci ispisani imenima, mora ih znati pretvarati u brojeve i lučiti moguće datume od nemogućih: 31. 4. 1983. ili 30. veljače 1600. Mora znati da 83. znači 1983', a ne neki daleki datum iz rimske povijesti.

Kada savlada sve zapreke, može početi s izračunavanjem broja proteklih dana između dva datuma, od kojih svaki može sezati u daleku prošlost. Treba uzeti u račun različit broj dana u mjesecima, dan više u prestupnim godinama (svake četvrte godine) i dan manje svake stoljetne godine (svakih 400 godina). A sve to zajedno vrijedi od 1582. godine, od uvođenja gregorijanskog kalendara, dok za ranije datume treba račun prilagoditi zakonima julijanskog kalendara.

Ostatak je jednostavan. Ali nije jednostavno sastaviti program za koji treba temeljito proučiti kalendarske zakone i provesti dva dana u strpljivom programiranju. Netko će to možda obaviti i brže. Meni će trebati sigurno više od dva dana.

Bavljenje jednim ovakvim programom prelazilo bi okvire ove knjige, pa zato prelazimo na jednostavniji zadatak pri kojem ćemo se služiti jednim nepoznatim programskim jezikom što sam ga nazvao FOODGOL[*], a cilj mu je prikazati temeljne principe programiranja.

---

[*] FOODGOL je autorova dosjetka skovana od riječi FOOD (hrana) i nastavka GOL koji nose imena nekih viših programskih jezika (ALGOL, COBOL itd.). Prim. prev.

Koraci, petlje i potprogrami

Pogledajmo sada program po kojem se čovjek hrani

Program objedovanja 1

1 uzmi nož i viljušku
2 odreži zalogaj
3 nabodi na viljušku
4 stavi u usta
5 odreži zalogaj
6 nabodi na viljušku
7 stavi u usta
8 odreži zalogaj
9 nabodi na viljušku
10 stavi u usta
11 odreži zalogaj
12 nabodi na viljušku

To je doduše prilično dosadno ali proces hranjenja teče upravo tako Međutim taj se program može načiniti jednostavnijim ako mu dodamo petlju koja ce isključiti dosadna ponavljanja

Program objedovanja 2

1 uzmi nož i viljušku
2 odreži zalogaj
3 nabodi na viljušku
4 stavi u usta
5 GOTO 2

I evo kako je to sada svaki put kad stavite zalogaj u usta program se vraca na programsku liniju 2 Ali što ce se dogoditi kada se tanjur isprazni?

Program objedovanje 3

1 uzmi noz i viljušku
1a Ima li još jela na tanjuru? Ako nema odloži nož i viljušku STOP
2 odreži zalogaj
3 nabodi na viljušku
4 stavi u usta
5 GOTO 1a

Ono što je nedostajalo bile je provjera kojom se testira da li treba radnju nestaviti ili ne To je sada izvedeno pomocu programske petlje i koraka 1a Proces hranjenja tece dalje sve dok u tanjuru ima hrane a kad je tanjur prazan program završava Pogledajmo još jednom prvu verziju programa Kako unaprijed znati koliko puta treba da ponovite ciklus stavljanja zalogaja u usta, deset ili sto sve dok viljuskom ne počnete strugati po praznom tanjuru Zbog toga postoji provjera koja tokom svakoga ciklusa kontrolira da li je tanjur prazan

Bilo bi dobro da se ovom skromnom jelovniku doda još i kolač U tom slučaju program se mora proširiti

Program objedovanje 4

1 uzmi nož i viljušku
1a Ima li još jela na tanjuru? Ako nema odloži nož i viljusku GOTO 8
2 odreži zalogaj
3 nabodi na viljusku
4 stavi u uste
5 GOTO 1a
6 uzmi kolač
7 Ima li još jela na tanjuru? Ako nema odloži nož i viljušku STOP
8 odreži zalogaj
9 nabodi na viljušku
10 stavi u usta
11 GOTO 7

Ova verzija programa promijenjena je na tri mjesta Prva se promjena odnosi na programsku liniju 1a kojom se naređuje uz uvjet da je prvo jelo dokrajčeno skok na liniju 6 u kojoj se traži kolač Nakon toga slijedi nova petlja koja se bavi jedenjem kolača opet s provjerom da li je kolač pojeden ili nije Ako jest program se prekida

Vidi se da su programske linije 8 9 i 10 točne kopije linija 2 3 i 4 a linije 11 i 5 su vrlo slične Stoga pokušajmo još jednom preurediti program kako bi ekonomičnije upravljao procesom hranjenja i to uvođenjem jednog podprograma

Program objadovanja 5

1 uzmi nož i viljušku
1a Ima li još jela na tanjuru? Ako nema odloži nož i viljusku GOTO 6
2a GOSUB 100
5 GOTO 1a
6 uzmi kolač
7 Ima li još jela na tanjuru? Ako neme odloži nož i viljušku, STOP
8 GOSUB 100
11 GOTO 7
100 odreži zalogaj
101 nabodi na viljušku
102 stavi u usta
103 RETURN

GOSUB je takoder komanda posuđena iz BASICa (vidi str 58-61) koja znači idi na naznačenu programsku liniju (u ovom slučaju 100) obavi što je temo naređeno i kad naiđeš na komandu RETURN vrati se na prvu liniju nakon mjesta s kojega si krenuo Naš program postupa upravo tako s linije 2a skače u potprogram na liniju 100 izvršava i vraca se na liniju 5 Kada provjera na 1a pokaže da je prvi dio programa obavljen preskače na 6 provjerava da li je tanjur pun pa ako je iznova skače na liniju 100 potprograma kako bi obavio konzumiranje kolača

Ovakav način programiranja štedi programski prostor kao u ovom slučaju gdje cijelim pro-

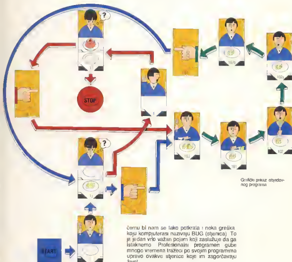
* GOTO naredba iz programskog jezika BASIC od engl go to — idi do' ili skoči na programsku liniju x Prim prev

cesom jedenja upravlja samo mali dio programa. Osim toga, jednostavnije su sve eventualne promjene u programu. Na primjer, ako ce netko umjesto viljuškom i nozem jesti prstima program se tome može lako prilagoditi. Ili budući da smo zaboravili nesretnika kojega hranimo narediti da žvaće i guta zalogaje (kad program stane on se može zagušiti hranom), pregledajmo još jednom naš program i usavršimo ga.

Program objedovanja 6.

```
1  uzmi nož i viljušku
1a Ima li još jela na tanjuru? Ako nema, odloži
   nož i viljušku. GOTO 6
2  GOSUB 100
5  GOTO 1a
6  uzmi kolač
7  Ima li jos hrane na tanjuru? Ako nema odloži noz i viljušku. STOP
8  GOSUB 100
11 GOTO 7
100 odreži zalogaj
101 nabodi na viljušku
102 stavi u usta
103 žvaći
104 gutaj
105 RETURN
```

Ovu smo promjenu lako obavili, no bilo bi mnogo teže da smo to isto pokušali izvesti u prvoj verziji programa. Na svake tri programske linije morali bismo dodavati po dvije nove, pri čemu bi nam se lako potkrala i neka greška, koju kompjuteraši nazivaju BUG (stjenica). To je jedan vrlo važan pojam koji zaslužuje da ga istaknemo. Profesionalni programeri gube mnogo vremena tražeci po svojim programima upravo ovakve stjenice koje im zagorčavaju život.



Grafički prikaz objedovnog programa

BASIC[*]

U prošlom smo poglavlju objasnili princip programiranja bez obzira na programski jezik Ustanovili smo da se naredbe ispisuju redom jedna po jedna a one koje su slične mogu se sažeti u jedinstveni potprogram na koji se može pozivati sa svakog mjesta glavnog programa Jednako je značajan i postupak testiranja koji provjerava izvršenje različitih stupnjeva glavnog programa

Istom programskom strukturom služi se mnoštvo programskih jezika kao FORTRAN ALGOL COBOL PASCAL i C U ovu grupu spada i BASIC programski jezik prvi na koji ce veci dio čitalaca ove knjige naići jer su mnogi njegovi dijalekti ugrađeni u većinu osobnih računala BASIC ima dakako svojih mane o kojima ćemo još raspravljati ali ime i mnogih prednosti pred drugim jezicima

Najveće njegova prednost je jednostavnost pogodna za početnika ali nije na odmet niti naprednijem Gotovo će svatko pri prvom susretu s kompjutorom moci nakon nekoliko minuta napisati neki jednostavni program ili se barem potpisati BASIC se odmah interpretira' što znači da se svaki dio programa moze odmah provjeriti da li radi Ostali viši programski jezici, moraju se najprije kompilirati odnosno prepustiti jednom posebnom programu, koji će napisani program temeljito preradili prije nego što ga stavi u pogon Taj proces može trajati koju minutu ali i nekoliko sati što početnika može zbuniti Ako BASIC program ne radi od prve (a to se najčešce i događa) može ga se vrlo jednostavno korigirati i pokušati nanovo

Treba još naglasiti da ne postoji standardni BASIC Na stranama 186-187 otisnut je popis instrukcija najćešće upotrebljavane verzije BASICa Savjetujem vam da se čitajući ovo poglavlje što češce navraćate na taj dio

BASIC se sastoji od tri vrsti instrukcija naredbe varijable i funkcije Najlakše je razumjeti naredbe to su glagoli PRINT (piši) naredba koja naređuje ispis nečega na ekranu, LPRINT, naređuje ispis na printeru + određuje da se dva broja imaju zbrojiti itd Varijable su vrlo slične algebarskim varijablama osim što se umjesto x i y u nekim dijalektima BASICa mogu stavljati slobodno izabrana imena kao PERO ili KUĆNI BROJ Postoje dvije vrsti varijabli prvom se predstavljaju brojevi a drugom tekst Ako sa u programu traži ime i godina rođenja korisnika tada to izgleda ovako

```
10 PRINT Molim napisite ime"
20 INPUT IME$
30 PRINT' godina rođenja"
40 INPUT GODINA
```

Ima nekoliko stvari koje treba uočiti u ovom malom programu Prvo svaka programska linija je numerirana i program ce izvršavati naredbe redom po brojevima Drugo dobra je navika da se brojevi povisuju po 10 (iako može i drugačije) tako da između starih linija ima mjesta za ubacivanje novih Nakon svake PRINT komande uvijek mora slijediti jedno ili više slova ili brojaka umetnutih između navodnika Takav niz zove se STRING a nakon naredbe INPUT računalo čeka da mu se preko tastature posalje nekakav podatak poslije kojega mora slijediti pritisak na tipku RETURN

Kada sa program pokrene (RUN) na ekranu ce se pojaviti ispis

```
Molim napišite ime
?
```

Nakon INPUTa uvijek se pojavljuje upitnik što znači da računalo nešto očekuje Treba mu udovoljiti ako želimo da nastavi redom i otipkati, recimo PERO a nakon toga RETURN BASIC sada zna da je vrijednost varijable IME$=PERO jer znak $ (dolar) nakon imena varijable služi zato da bi BASIC znao što treba tretirati kao tekst, a što kao brojeve Procesor drugačije postupa s brojevima a sasvim drugačije s tekstom

No vratimo se našem programu Pošto smo otipkali svoje ime na ekranu ce se pojaviti slijedeći ispis

```
godina rođenja
?
```

Računalo opet čeka na odgovor Tada mu otipkate recimo 1962 i opet RETURN Taj broj je sada postao vrijednost varijable koju kompjutor prepoznaje pod imenom GODINA' Budući da je BASIC nastao kao matematički jezik on sve varijable koje nisu označene oznakom $ shvaca kao brojeve

No kako se s programima treba ekonomizirati jer pred njima obično stoji veliki posao BASIC predviđa sažetiju metodu programiranja tako da cijeli naš program može stati u jednu programsku liniju

```
10 INPUT 'Molim unesite vaše ime i godinu rođenja odvojene zarezom IME$ GODINA
```

Na ekranu se sada pojavljuje string napisan unutar navodnika poslije INPUTa Nakon stringa je znak koji BASICu kaže da ne pomiče kursor u novi red što bi se inače odmah dogodilo da ovoga znaka nema Sada on očekuje da mu se otipka

```
PERO 1962
```

Treba odmah naglasiti da se ta dva podatka svakako moraju odvojiti upravo sa zarezom a ne s točkom ili crticom BASIC u ovom slučaju

[*] BASIC je akronim za Beginners All-purpose Symbolic Instruction Code — dijaloški orijentiran viši programski jezik koji nema nikakve veze sa pojmom Basic English pojednostavnjenim engleskim jezikom skučenoga rječnika Prim prev

ne rezumije nikakav drugi znak osim zareza Ako na to ne pazimo kompjutor će se odmah poželiti ispisom ERROR MESSAGE' (dojave pogreške)

Vratimo se programu Uveli smo sada dvije varijable jednu numeričku i jednu string varijeblu (ili alfanumeričku) S tim se varijablama može sada nesto započeti ali kako se radi o dvjeme različitim vrstima varijabli postoje i dvije različite vrsti operacija koje se mogu s njima izvoditi

Pogledajmo najprije operacije sa stringovima Možda bi bilo zgodno da se kompjutor javi pozdravom osobi koja mu je upravo saopcila svoje ime Možemo mu narediti da izjevi

Dobro jutro Pero! Drago mi je što smo se upoznali!

Ime nekoliko načina na koje se taj trik može izvesti Jednostavno treba cijelu poruku ispisati ne ekranu pomoću ovakvog programa

```
20 ?"Dobro jutro
30 ?IME$
40 ? !Drago mi je što smo se upoznali!
```

Mnogi BASICi priznaju znak ? kao skraćenicu za naredbu PRINT sto olakšava tipkanje I ponovimo još jednom da znak sprečava prelazak u novi red nakon svake programske linije Bez njega bi ispis izgledao ovako

Dobro jutro
Pero
!Drago mi je što smo se upoznali!

Drugi način da povežemo tri stringe u jednu cjelinu je slijedeci

```
20 B$= 'Dobro jutro, +IME$+"! Drago mi je
što smo se upoznali!
30 ?B$
```

Uveli smo novu varijablu B$ Znak + upotrijebljen između dva stringa znači da kompjutor treba jednostavno 'zalijepiti jednog uz drugog Može se također primijetiti da je znak '= primijenjen neobično barem sa stajališta školske algebre Nezgodno je što se BASIC služi znakom = u dva različite smisla što može zbuniti početnika

Znak jednakosti kako je ovdje upotrijebljen znači stavi ovo sto je s lijeve strane znaka na desnu stranu, ili 'izjednači ovo što je s lijeve strane znaka s onim s desne (Neki jezici upotrebljavaju kombinaciju = za ovu vrst jednakosti kako bi se razlikovala od prave)

IF A=B THEN (ako je A=B tada)

Za pokus možemo to načiniti i u našem programu Možda susjeda Ivana ne volimo previše pa možemo uvesti novu liniju (evo primjera kako je konsna navika da se brojevi linija povecavaju po 10)

```
15 IF IME$="IVAN THEN PRINT udri
ga" GOTO 10
```

To je prava jednakost i služi za provjeru da li se radi o nesimpatičnom Ivanu ili ne Naredba IF znači 'ako Dakle ako je IME$ Ivan, tada program izvršava naredbu koje slijedi nakon THEN (tada) Ako pak IME$ nije Ivan vec Marko program će kao ranije ignorirati ovu liniju i prijeći na izvršenje linije 20

Ovdje smo upoznali jedan novi simbol On dozvoljava da se nakon izvršenja naredbe PRINT UDRI GA" doda u istoj liniji još jedna GOTO 10 Ako je uvjet zadovoljen program skače na liniju 10 a ako nije nastavit će starim putem na liniju 20

Matrica

Ne možemo daleko odmaći s BASICom (niti s ikojim drugim programskim jezikom) a da se ne susretnemo s pojmom matrice (array) To je varijable što se sastoji od više praznih polja, u koja se umjesto jednoga može smjestiti više stringova ili brojeva Najjednostavnija vrst matrice je one koju nazivamo vektor', samo s jednim radom polja.

Pretpostavimo da je predmet našeg zanimanja meteorologija i treba nam program kojim cemo bilježiti količinu padavine svakog dana u tjednu Na početku programa moramo rezervirati prostor ili dimenzionirati matricu jer BASIC to izričito traži

```
DIM KIŠA(7)
```

Prema ovoj uputi program zna da namjeravamo raditi s varijablom KIŠA' od 7 polja Svakom polju može se posebno pristupiti pozivajuci se na brojeve polja KIŠA(1) KIŠA(2) itd Sada možemo napisati mali program koji ce počevši od nedjelje bilježiti padavine registrirane svakog dana tokom tjedna

```
10 DIM KIŠA(7)
20 FOR K=1 TO 7
30 INPUT' današnja količine kiše' KIŠA(K)
40 NEXT K
```

Ovaj se program služi petljom koja se vrti između linija 20 i 40 Petlja počinje postavljanjem varijable K na vrijednost 1 i raste svaki put za 1 sve dok ne dosegne vrijednost 7 U svakom okretaju petlje program čeka da se upiše danesnja količina kiše' i korisnik utipkava podatke od kojih ce svaki biti upisan u jedno polje matrice KIŠA

Na priloženoj slici vidi se da je ponedjeljak bio vlažan a da je u subotu i nedelju pljuštalo kao iz kabla

	1	2	3	4	5	6	7
KIŠA	9	01	04	6	1 2	4	2
DAN	PON	UT	SR	ČET	PET	SUB	NED

KIŠA	1	9	01	04	6	1 2	4	2
T MAX	2	14	22	24	20	17	16	18
T MIN	3	3	9	10	8	7	[7]	7
VLAGA	4	7	6	5	55	6	8	7
SUNCE	5	1	8	9	5	1	0	0
VJETAR	6	10	2	2	15	10	25	20

U BASICu se može formirati i string-matrica pomoću koje se pravi zgodan program za upisivanje padavina svakoga dane u tjednu Prvo trebe dimenzionirati string-matricu pod imenom OANI u koju ćemo pohraniti skraćena imena dana pon ut sr ned

```
10 DIM KIŠA(7),OANI$(7)
20 DATA PON, UT SR ČET PET SUB NED
30 FOR K=1 TO 7
40 READ DANI$(K)
50 NEXT K
60 FOR K=1 TO 7
70 PRINT "padavine u" OANI$(K) INPUT
KIŠA(K)
80 NEXT K
```

Prva petlja počinje na liniji 30 e zatvara se sa NEXT K na liniji 50 i sadrži (linije 20) imena dana upisanih u obliku podataka (DATA) odvojenih zarezom Naredba READ (čitej) u liniji 40 umeće redom po jedan podatak u rubriku OANI$(K)

Druge petlja (linija 60—80) ispisuje na ekranu pitanje padavine u' dodaje ime dana pa cijelo pitanje glasi padavine u pon ? I čeka odgovor Tako je formirana dvodimenzionalna matrica od 7 × 7 polje (Postoje dakako još svakojake matrice koje mogu sadrzavati više redova numeričkih ili string-varijabli)

Na nekoj meteorološkoj postaji na primjer može se svaki dan upisivati gomila numeričkih podatake koji će govoriti o pedavinema maksimalnim i minimalnim temperaturama vlažnosti zraka brzini vjetra i broju sunčenih sati Sve se to (7 dana po 6 podataka) mora na početku programa dimenzionirati naredbom

```
10 DIM METEOROLOGIJA(7 6)
```

Programski pristup je isti kao u već spomenutom primjeru Ne primjer podatak o temperaturi u petak nalazi se na METEOROLOGIJA(7 3) i iznosi 7

Brojevi

Vidjeli smo što sve BASIC moža uraditi sa stringovima a sada da posvetimo pažnju brojevima Recimo da treba upisati svoje ime i prihod i pretvoriti ta podetke u verijeble Varijabla DOHODAK sadrževat će numeričke varijable s kojima se mogu izvoditi aritmetičke operacije

```
10 INPUT" Molim upišite ime i placu odvojene
zarezom IME$ PLAĆA
15 IF IME$="IVAN THEN PRINT"udn
ga!" GOTO 10
20 B$= Dobro jutro + IME$ + "drago mi je sto
smo se upoznali
30 PRINT B$
40 NOVA=PLAĆA*8
50 PLAĆA=PLAĆA-NOVA
60 PRINT NOVA neka bude moje"
PLAĆA neka bude tvoje"
```

Sada smo dodali tri nove programske linije 30 40 i 50 koje mogu djetovati neobicno Linija 40 izračunava novu varijablu dali smo joj ime NOVA vrijednost koje iznosi 80% verijable PLAĆA (množeći je s 0 8) Simbol '*' je znak za operaciju množenja jer × bi se moglo zamijeniti s nekom drugom varijablom

Linija 50 ce postati jasna ako se podsjetimo da simbol '=' znači stavi ono što je s desne strene znake na lijevu pe zato ova linije kaže izjednači novu vrijednost varijable PLAĆA s vrijednošću stare umenjenu za verijablu NOVA' Linija 60 ispisuje ne ekranu kompjutorovo zabrinuto mišljenje o tekućoj ekonomskoj situeciji

1200 neka bude moje 2800 neka bude tvoje

Postoje još mnoge operacije sto ih BASIC izvodi s brojevima Može ih oduzimati i dijeliti potencirati logaritmirati i antilogaritmirati izračunavati sinuse i kosinuse a pomoću njih i sve ostale trigonometrijske funkcije Mogu se generirati slučajni (RANDOM) brojevi Doduše ne baš sasvim 'slučajni jer kompjutor kao savršeno logička nepreva ne može izbacivati prave slučajne brojeve što će reći odabirati brojeve na čistu sreću Oni se generiraju jednom dosta kompliciranom procedurom ali se površnom promatraču zaista to čini kao plod slučajnosti (vidi str 183)

Da bi se, na primjer, izračunao logaritam broja A treba jednostavno otipkati

```
70 B=LOG(A) PRINT B
```

BASIC ima mnogo funkcija i LOG je jedna od njih Neke funkcije mogu se primijeniti i na stringove

```
10 B$= "KOMPJUTOR U KUĆI"
20 PRINT LEN(B$)
```

U liniji 10 nalazi se riječ o kojoj želimo saznati od koliko slove se sastoji Odgovor na to pitanje daje linija 20 16 slova (14 slova i dva razmaka)

Grafike

U posljednje vrijeme pojavljuje se sve više malih kompjutora s ekranima visoke rezolucije ali

(prema CAD standardu vidi str 102—107) skromnih grafičkih mogućnosti Najjednostavnije varzije dozvoljavaju da se na ekranu nacrta točka povuče ravna linija i da se neka površina ispuni bojom Boja se definira brojem Na primjer da se nacrta točka na poziciji X Y (koordinate se računaju od donjeg lijevog ugla) treba otipketi

PLOT POINT (C.X Y)

Ako pak želimo povući crtu boje 5 od točke 34 56 do točke121 444, naredit ćemo

PLOT LINE(5 34 56 121,444)

Slijedeće naredba ispunit će bojom broj 3 pravokutnik definiran koordinatama točake njegova donjeg lijevog i gornjeg desnog ugla

PLOT FILL(3 34 56 121 444)

De bi se pod kontrolom programe povukla crta treba najprije odrediti na koje ce mjesto pasti točke Ne primjer za crtu pod kutom od 45 stupnjeva treba napisati

```
100 FOR K  1 TO 4
110 PRINT AT X  K Y  K
120 NEXT K
```

Na ekranu je efekt slijedeći

```
•
 •
  •
   •
    •
```

Ima dosta računala koja uopće nemaju grafičkih mogućnosti pogolovo ona gređena za poslovne svrhe U takvim slučajevima nije jednostavno postaviti neki znak ne X Y poziciju ekrana Svako računalo ima naredbu "HOME" (kući) kojom se kursor šalje na svoju ishodišnu točku Tada ga trebe pomicati najprije po X e onde po Y osi i kada dođe na željeno mjesto otipketi znek Postupak trebe ponoviti za svako novo slovo

Sve to ne zvuči ohrabrujuće za vaše buduce planove de u novoj kompjutorskoj igri nacrtate svemirsku postaju na planeti Hyperon Dakako najbolje bi bilo da se poslužite digitalizecijskom pločom (digitizing pad vidi str 110—111) — ako je imate A bit ce da je nemate pa onda evo načine kako se može crtati uz pomoc BASICa i DATA naredbi

```
10 HOME$  CHR$(29) REM YOUR HOME
   CHARACTER
20 DOWN$  CHR$(10) REM CURSOR DOWN
30 RTS  CHR$(32) REM CURSOR RIGHT
40 W  79 H  23 REM WIDTH AND HEIGHT OF
   YOUR SCREEN
200 INPUT  POSITION  X Y
210 INPUT  WHICH SHIP TO PRINT  I
220 ON I GOSUB 5000 6000 7000
230 A$  INPUT$(1)
240 REM ·······················
900 READ HT
910 IF Y  HT>H THEN HT  H  Y
920 FOR I  1 TO HT
930 READ D$
1000 PRINT HOME$
1010 PRINT STRING$ (Y  I DOWN$)
1020 PRINT STRING$ (X RTS)
1030 PRINT LEFT$ (D$ W  X)
1035 NEXT I
1040 RETURN
4998 REM
4999 REM ·······························
5000 RESTORE 10000 GOSUB 900 RETURN
6000 REM JUMP TO ANOTHER DRAWING
10000 DATA 7
10010 DATA          •       •
10020 DATA          •       •
10030 DATA          • ••••••••
10040 DATA          • ••••••••
10050 DATA     ••••••••••••••••••
10060 DATA      •••••••••••••••••
10070 DATA       ••••••••••••••••
```

Slika je nacrtana točno onako kakva je u programu uz DATA neredba (linije 10010-10070) U liniji 10000 upisan je broj koji progremu unaprijed saopcava s koliko linija treba sliku nacrtati U linijame 10—50 određuje se veličina ekrana

U liniji 200 traži se pozicije slike na ekranu a slijedece ostavljaju prostor za druge slike koje eventualno treba nacrtati Naredbe ON GOSUB selje izvršenje programa na prvi drugi ili treći potprogram od kojih je ovdje samo prvi specificiran a oslali se mogu dodati

Potprogram 5000 upucuje DATA mehanizam na citanje od linije 1000 Ako na liniji 11000 ime jos jedne slika tada u potprogram 6000 treba upisati iste naredbe kao u liniji 5000

RESTORE 11000 GOSUB 900 RETURN

Program teda opet skace na liniju 900 Na ovoj i na slijedećoj liniji izračunava koliki dio slike mora izostaviti ako je pozicionirana preblizu ruba ekrana Petlja između linije 920 i 1040 iscrtava svaku DATA liniju na određenoj poziciji ekrana

Ako bacite pogled na strane 186—187 vidjet cete da tamo ima mnogo naredeba i funkcija BASICa kojih se ovdje nismo dotekli Međutim ako zaista želite neučiti BASIC onda vam neme druge nego da kupite neki mali kompjutor te da počnete raditi na njemu Naucitl programirati bez kompjutora isto je sto i učiti upravljati biciklom — bez bicikla Tokom učenja BASICa također ćete shvatiti (a to je najteže) onaj navjerojatno glup način na koji misli kompjutor A kada i to shvatite moći ćete ocijeniti što ovi strojevi mogu a što ne

Citirani programi odabrani su iz časopisa Practical Computing. Zahvaljujem izdavaču (IPC Electrical and Electronic Press Ltd) što ja dozvolio da ih objavim u ovoj knjizi.

Nastojao sam odabrati što različitije programe koji će najbolje ilustrirati osnovne programske tehnike a da ujedno izbjegnem postavljanje prevelikih zahtjeva na računalo i programera.

Ne početku većine citiranih programa trebal će dodati naredbu za čišćenje ekrana (clear screen).

Šifriranje

Budući da je povijest kompjutora usko vezana uz probleme probijanja neprijateljskih šifri tokom prošlog rata red je da pokažemo meli program koji šifrira i dešifrira poruke ali bez bojazni da će netko probiti tajnu državne šifre.

Postupak šifriranja je u osnovi jednostavan treba poruku slovo po slovo pretvoriti u brojeve. ASCII kôdovi mogu sjajno poslužiti u tu svrhu. Zatim treba svaki dobijeni broj zamijeniti nekim drugim brojem ili na način koji je poznat samo vama i vašem korespondentu. Najjednostavnije je da se unaprijed pripremi niz slučajnih (random) brojeva i dodaju im se određeni brojevi slova. To je shema po kojoj su pravo tzv. jednokratne šifre koje su u pravilu neprobojne. Slaba strana jednokratne šifre je u tome što niz slučajnih brojeva mora biti jednako dugačak kao i sama poruka. U drugom svjetskom ratu su kriptolozi probijali šifre mukotrpnim uspoređivanjem svake ma i najmanje razumljive mrvice informacije što su je uspjeli iščačkati iz gomile signala s onim što su već znali ili naslućivali.

Da bi osujetile svaku mogućnost takve analize modeme armije i diplomatske službe moraju svakodnevno razašiljati gomile jednokratnih šifri s kojima se šifriraju ogromne količine materijala. To dakako nije praktično, pa se tražio način generiranja niza slučajnih brojeva koji su toliko slučajni da ih neprijateljska strana ne može predvidjeti. To nije lako jer svaki mehanički postupak, pne ili kasnije počinje iznova generirati isti niz brojeva. Tada neprijateljski kriptolog ima šansu da probije šifru a pogotovo ako se stanu pojavljivati neke pravilnosti šanse za probijanje su još veće.

Jednostavni način generiranja niza brojeve (ključa) koji je dosta slučajan da ga ne bi lako otkrili nepozvani e ipak dovoljno jednostavan da ga pamte upućeni je onaj čija je osnova tzv. 'ključna riječ'. Svako slovo ključne riječi služi kao šifra odgovarajućeg slova poruke i ponavlja se ako je poruka dulja od nje. To objavlja slijedeći program.

```
10 '*****CODE2******
20 KEY$="ZEBRA"
30 INPUT"String to be coded";I$
40 FOR K=1 TO LEN(I$)
50 L=L+1
60 IF L>LEN(KEY$) THEN L=1
70 K$=MID$(KEY$,L,1)'GET THE L'TH LETTER
 OF KEY$
80 J$=MID$(I$,K,1)
90 O=ASC(K$) XOR ASC(J$)
100 PRINT O;" ";
110 NEXT K
```

Dešifriranje vrši program DECODE 2

```
10 '*****DECODE2******
20 KEY$="ZEBRA"
25 INPUT "Length of code";LC
40 FOR K=1 TO LC
50 L=L+1
60 IF L>LEN(KEY$) THEN L=1
70 K$=MID$(KEY$,L,1)'GET THE L'TH LETTER
 OF KEY$
80 INPUT"Next code number";J
85 LPRINT J;" ";
90 O=ASC(K$) XOR J
100 LPRINT CHR$(O);
110 NEXT K
```

Ozbiljnom kriptologu takva jednostavna shema ne zadaje mnogo glavobolje jer se ključ ponavlja ovisno o duljini ključne riječi.

Ponavljanje se može odgoditi ako se izlaz još jednom kodira novom ključnom riječi i u tom slučaju će nastupiti tek nakon što se iscrpio umnožak broja slova obje riječi. Postupak se može ponoviti se proizvoljnim brojem ključnih riječi na primjer šest riječi svake po osam slova osigurat će kodiranje bez ponavljanja za 8^6 — a to je više od 250000 slova. Šifra je sigurna ako se ključne riječ promijeni prije nego što bude odaslan ovoliki broj slova.

Anagram

U ovom programu akcent je na manipulaciji teksta a to je ono što kompjutor najlakše radi. Anagram ili premetaljka je vrst zagonetke s kojom će se svatko kad-tad pozabaviti. Program prima niz slova što ih treba ispremetati pita da li je koje slovo poznato i ispisuje sve mogućnosti. U ovom primjeru zadana su slova AARAS koje treba anagramirati a slova —N—G—M— su poznata. Rezultat je otisnut desno.

```
100 'ANAGRAM PROGRAM
110 'M G PRITCHARD, PC APRIL 1980
120 PRINT"ANAGRAM"
130 PRINT"----------"
140 INPUT"Type only the letters to be re
arranged";A$
150  PRINT
160 L=LEN(A$)
170 INPUT"Are there any letters known (Y
/N)";Q$
180 IF Q$="N" OR Q$="n" THEN 240
190 IF Q$<>"Y" AND Q$<>"y" THEN 170
200 PRINT:PRINT:INPUT"Type the known let
ters, eg `-B--D-`";K$:W=L
201 FOR J=1 TO LEN(K$)
203 IF MID$(K$,J,1)="-" THEN T=T+1
205 IF LEN(K$)<>L THEN PRINT"Wrong lengt
h; try again":GOTO 200
206 NEXT J
```

```
ANAGRAMS  ANAGRRMA
ANRGASMA  ANAGASMR
ANSGARMA  ANSGAAMR
ANAGASMR  ANRGASMA
ANAGARMS  ANAGSAMR
ANSGRAMA  RNAGASMA
RNAGAAMS  RNAGSAMA
RNAGASMA  RNSGAANA
ANAGARMS  ANAGSANR
ANAGRSMA  ANRGASMA
ANSGAANR  ANSGRAMA
SNAGRAMA  SNAGAAMR
SNRGAAMA  SNRGAAMA
SNAGARMA  ANAGSAMR
ANAGRSMA  ANAGRAMS
ANRGSAMA  ANSGARMA
ANSGAAMR  ANRGAAMS
ANAGSRMA  ANAGSRMA
ANAGASMR  RNAGAAMS
ANSGAANR  RNAGSAMA
RNAGASMA  RNSGAAMA
RNAGSAMA  ANAGSRMA
ANAGASMR  ANRGAAMS
ANAGSAMR  ANSGRAMA
ANSGARMA  SNAGRANA
SNAGAAMR  SNRGAANA
SNRGAAMA  ANRGAAMS
SNAGRAMA  ANAGSRMA
ANAGASMR  ANAGARMS
ANAGARMS  ANRGSAMA
ANSGRANA  ANSGRAMA
ANRGAAMS  RNAGSAMA
ANAGRSMA  RNAGAAMS
RNAGAAMS  RNSGAAMA
RNAGASMA  ANAGRANS
RNSGAAMA  ANRGSAMA
ANAGRSMA  SNAGRAMA
ANRGAANS  SNAGARMA
ANSGARMA  SNAGAAMA
SNAGARMA  ANRGASMA
SNRGAANA  ANAGSAMR
ANAGARMS  ANAGRSMA
ANRGSANA  ANRGSAMA
ANSGRAMA  ANSGAAMR
ANAGSAMR  RNAGSAMA
ANAGRAMS  RNAGASMA
ANSGARMA  RNSGAAMA
RNAGAAMS  ANAGASMR
RNAGSAMA  ANRGSAMA
ANAGRAMS  SNAGAAMR
ANAGSRMA  SNAGAANR
ANSGAAMR  SNAGARMA
RNAGARMA  ANRGAAMS
SNRGAAMA  ANAGRSMA
SNAGRAMA  ANAGRAMS
ANAGARMS  ANRGASMA
ANRGASMA  ANSGARMA
SNAGARMA  RNAGASMA
SNAGRAMA  RNAGAANS
SNAGAAMR  RNSGAAMA
```

```
210 T=0:FOR J=1 TO LEN(K$)
212 IF MID$(K$,J,I)="-" THEN T=T+1
215 NEXT J
230 GOTO 270
240 K$="":FOR J=1 TO L:K$=R$+"-":NEXT J
250 PRINT:INPUT"Number of letters to beg
in",W
260 IF W<1 OR W>L OR W<>INT(W) THEN PRIN
T"Error":GOTO 250
270 DIM B$(L),C$(L),Q(L)
280 PRINT:PRINT
290 GOSUB 5000
300 FOR J=W TO L
310 R=1
320 Q(R)=1
330 IF B$(Q(R))="" THEN 440
340 C$(K)=B$(Q(R)):B$(Q(K))=""
350 K=R+1
360 IF R<=J THEN 320
370 A=1
380 FOR S=1 TO LEN(K$)
390 IF MID$(R$,S,1)="-" THEN PRINT C$(A)
;:A=A+1:GOTO 410
400 PRINT MID$(K$,S,1);
410 NEXT S:PRINT,
420 R=J
430 B$(Q(K))=MID$(A$,Q(R),1)
440 Q(K)=Q(R)+1
450 IF Q(K)<=L THEN 330
460 K=K-1
470 IF K>=1 THEN 430
480 NEXT J
490 END
5000 FOR N=1 TO L
5010 B$(N)=MID$(A$,N,1)
5020 NEXT N
5030 RETURN
```

Cloze[1]

Program je nastao kao rezultat istraživanja u vježbenju engleskog jezika[2]. Programske linije 100—200 primaju dio nekog teksta pa se može, recimo, napisati

WHEN I CAME BACK FROM MY HOLLIDAY I RAN STRAIGHT OUT INTO THE GARDEN TO SEE MY RABBIT WILLIAM[3]

Program tada pita koliko riječi treba ispisati da bi nakon njih, a umjesto onih skrivenih, napisao crtice (—). Ako nastavnik zatraži dvije, očistit će ekran i napisati

WHEN I ——— BACK FROM —— HOLLIDAY I ——— STRAIGHT OUT ——— THE GARDEN —— SEE MY ——— WILLIAM

Učenik sada treba umetnuti riječi, jednu po jednu, koje nedostaju. Ako prva riječ odgovara rečenice će se ispisati iznova do mjesta gdje nedostaje sljedeća riječ.

S tehničke strane najzanimljiviji dio su linije 100-200 u koje se ubacuju skrivene riječi. Slova se pojedinačno primaju s tastature naredbom I$=INPUT$(1) i dodaju tekućoj riječi u liniji 180. To znači da se programer morao baviti brisanjem (linija 130), ali i stalnim testiranjem znaka za svršetak, u ovom slučaju •. U još savršenijoj izvedbi, koja bi ekran potpuno kontrolirala kursorom, mogla bi se koristiti slična vrst ulaza. Na primjer, da pomakne kursor dolje, korisnik bi mogao utipkati CONTROL D (↑D). Čim bi ulaz prepoznao ASCII 4, otišao bi u potprogram i napisao CHR$(8) koji pomiče kursor za jedan red dolje. Linija 310 također bi se mogla poboljšati kad bi se uvela mala i velika slova.

Bilo bi vrijedno pozabaviti se linijama 270-277 i ustanoviti kako taj program na zgodan način ispisuje crtice na mjestima gdje manjkaju skrivene riječi.

```
10 GOSUB 1000
15 M=0
20 PRINT "CLOZE"
30 'REM   BY CHRIS HARRISON, PC JUNE 198
2
40 FOR I=1 TO 1000:NEXT
49 'REM A$ = WORDS OF TEXT; R$= STUDENT'
S RESPONSE TO FIRST AVAILABLE BLANK
50 DIM A$(200),  R$(200)
59 'REM ASTERISK REQUIRED TO SHOW END
60 PRINT"WRITE YOUR TEXT IN HERE WITHOUT
 COMMAS. WHEN YOU HAVE FINISHED, ENTER A
 SPACE AND AN ASTERISK.
80 M=M+1 N=1'REM N SETS LETTERS IN THE W
ORD COUNTER
90 'REM TEXT INPUT ONE LETTER AT A TIME
100 I$=INPUT$(1) 'REM EQUIVALENT TO INKE
Y$ OR GET$
120 'REM NOW WE ALLOW FOR ERASURE. IF N<
1 WE START THE WORD AGAIN IF BACKSPACE I
S USED THEN REVERSE THE CURSOR AND IGNOR
E THE PREVIOUS LETTER.
130 IF N<1 THEN 90 ELSE IF I$=CHR$(8) TH
EN A$(M)=LEFT$(A$(M),N-2)):N=N-1:PRINT C
HR$(8) GOTO 100
140 IF I$="" THEN 100
149 'REM SPACE NOT COUNTED AS A WORD
150 N=N+1.IF I$=" " THEN PRINT" ";·GOTO
80
159 'REM IF I$="*" THEN END TEXT
160 IF I$="*" THEN 210
169 'REM IF RETURN KEY HIT NO ACTION TAK
EN
170 IF I$ = CHR$(13) THEN 100
179 'REM EACH WORD BUILT UP FROM LETTERS
180 A$(M)=A$(M)+I$
189 'REM NOW PRINT OUT EACH LETTER
190 PRINT I$;
199 'REM REPEAT THE PROCESS
200 GOTO 100
209 'REM S IS USED FOR SPACING
210 PRINT.PRINT:PRINT·INPUT" WHAT INTERV
AL DO YOU WANT";S
215 'REM  AN ENTRY OF 3 MEANS 2 WORDS AN
D THEN A BLANK
220 S=S+1
229 'REM NOW WE'RE READY  CLEAR SCREEN O
N LINE 240
240 GOSUB 1000
269 'REM NOW PRINT A SPACE BEFORE EACH W
ORD ,THEN THE WORDS EXCEPT FOR THOSE EVE
RY S-1 WHEN WE PRINT AS MANY DASHES AS T
HERE ARE LETTERS
270 PRINT" "; FOR I=1 TO M STEP S
272 FOR J=I TO I+S-1
```

[1] Postupak testiranja shvaćanja pročitanog teksta pomoću riječi koje sistematski nedostaju. Op. prev.

[2] Pomoću istog programa može se vježbati svaki drugi jezik a ne samo engleski. Op. prev.

[3] Čim sam se vratio s ferija, istrčao sam ravno u vrt da vidim svog zeku Williama. Op. prev.

```
275 PRINT A$(J);" ";:NEXT J
277 PRINT STRING$(LEN(A$(J)),"-");" ";·I
=I+1·NEXT I
279 'REM NOW ALLOW FOR AS MANY ANSWERS A
S THERE ARE BLANKS
280 FOR I=S+1 TO M STEP S+1
299 'REM NOW BLANK OUT THE FIRST PART OF
 THE SCREEN - WE WILL USE FOR MESSAGES A
ND ALLOW FOR ANSWERS
300 PRINT"NOW FILL IN THE BLANKS";:INPUT
 R$(I)
309 'REM IS THE ANSWER CORRECT?
310 IF R$(I)=A$(I) THEN 350
325 IF V>1 THEN PRINT"BAD LUCK. THE WORD
 IS ";A$(I); ELSE 330
326 FOR L=1 TO 300:NEXT L:GOTO 350
329 'REM IF WRONG REPLY, GIVE ANOTHER TR
Y
330 PRINT"NO, BAD LUCK, TRY AGAIN";V=V+1
:GOTO 300
349 IF THE REPLY CORRECT, REPRINT THE OR
IGINAL TEXT UP TO THE POINT REACHED
350 FOR K=1 TO I·PRINT A$(K);" ";·NEXT K
:PRINT
370 V=0 NEXT I
380 PRINT"THE  WHOLE SENTENCE WAS:" FOR
I=1 TO M.PRINT A$(I);' '; NEXT I·PRINT
389 'REM ALLOW FOR A RERUN
390 PRINT"ANOTHER GO? (Y/N)"; INPUT R$
410 PRINT"FINISH"
1000 'REM CRUDE WAY OF CLEARING SCREEN
1010 FOR K=1 TO 24·PRINT NEXT K
1020 RETURN
```

Život

Kompjutor ne može učiniti više nego čovjek s olovkom i papirom dakako ako ima dovoljno vremena. Ali kompjutor može sve obaviti tako brzo da se koji put čini kao da ima neki svoj vlastiti život. Prekrasan primjer za ovu misao je igra Život autor koje je britanski matematičar John Conway. To nije igra u običnom smislu te riječi jer kada jednom krene ona igra sama od sebe ali zato nija ništa manja fascinantna od drugih.

Ideja je vrlo jednostavna. Ekran je podijeljen na kvadrate koji su u kompjutoru sadržani kao memorijski bitovi ili bajtovi. Svaki kvadrat može biti prazan ali može sadržavati i živu ćeliju — 1 ili 0. Svaka ćelija ima 8 susjednih ćelija

```
0  0  0
0  ×  0
0  0  0
```

U svakom krugu kompjutor ispituje kvadrate sa ćelijama držeći se slijedećih pravila

mrtva ćelija oživljava ako ima tri živa susjeda (ćelije su trospolne a ne dvospolne)

ćelije koja imaju tri ili više živih susjeda umiru od iscrpljenosti

ćelija s jednim ili nijednim živim susjedom umire od usamljenosti

Na igraču je da postavi nekakav početni raspored ćelija i da starta igru. Nakon toga ostaje mu samo da s divljenjem promatra kako se razvija 'život'

```
10 W=40:R=24:CLR$=CHR$(29)'REM YOUR SCRE
EN WIDTH, HEIGHT AND HOME
15 W=W 1:H=H-1
20 DIM A(W+2,H+2),B(W+2,H+2)
30 'REM INPUT START PATTERN ENDED BY ""
40 PRINT"Input lines of cells. Hit RETUR
N alone to finish"
45 L=0
50 INPUT A$:L=L+1
60 IF A$="" THEN 200
65 IF ML<LEN(A$) THEN ML=LEN(A$)'GET LON
GEST LINE
70 IF LEN(A$)>W THEN PRINT"too long"·GOT
O 50
80 IF L>H THEN 200
90 FOR K=1 TO LEN(A$)
100 C$= MID$(A$,K,1):IF C$<>" " THEN A(K
,L)=1
110 NEXT K
115 GOTO 50
200 'REM, NOW SHIFT START PATTERN TO MID
DLE OF SCREEN
210 SW=INT((W-ML-1)/2)
220 SH=INT(H-L-1)/2
230 FOR K=1 TO ML
240 FOR J=1 TO L
250 B(K+SW,J+SH)=A(K,J)
260 NEXT J
270 NEXT K
280 GOSUB 10000
300 'REM NOW DO A GENERATION FROM B TO A
310 POP=0
320 FOR K=1 TO H
330 FOR J=1 TO W
340 N=0.C=B(J,K)'REM ZERO NEIGHBOUR COUN
T, SAVE CELL VALUE
370 N=B(J-1,K-1)+B(J,K-1)+B(J+1,K-1)+B(J
-1,K)+B(J+1,K)+B(J-1,K+1)+B(J,K+1)+B(J+1
,K+1)'REM COUNT UP NEIGHBOURS
410 GOSUB 30000
420 A(J,K)=NXT:POP=POP+1'REM PUT NEXT GE
NERATION INTO A
430 NEXT J
440 NEXT K
450 IF POP=0 THEN STOP
460 GOSUB 20000
490 'REM NOW PUT GENERATION INTO B
500 POP=0
510 FOR K=1 TO H
520 FOR J=1 TO W
530 N=0, C=B(J,K)
540 N=A(J-1,K-1)+A(J,K-1)+A(J+1,K-1)+A(J
-1,K)+A(J+1,K)+A(J-1,K+1)+A(J,K+1)+A(J+1
,K+1)
550 GOSUB 30000
555 B(J,K)=NXT:POP=POP+1
560 NEXT J
570 NEXT K
580 IF POP=0 THEN STOP
590 GOSUB 10000
600 GOTO 300
10000 'REM SUBROUTINE TO DISPLAY ARRAY B
10005 PRINT CLR$;
10010 FOR K=1 TO H
10020 FOR J=1 TO W
10030 IF B(J,K)=1 THEN PRINT"*"; ELSE PR
INT" ";
10035 'PRINT B(K,J);
10040 NEXT J
10050 PRINT
10060 NEXT K
10070 RETURN
20000 'REM SUBROUTINE TO PRINT ARRAY A
20005 PRINT CLR$;
20010 FOR K=1 TO H
20020 FOR J=1 TO W
20030 IF A(J,K)=1 THEN PRINT"*"; ELSE PR
INT" ";
20035 'PRINT A(J,K);
20040 NEXT J
```

```
20050 PRINT
20060 NEXT X
20070 RETURN
30000 'REM DO DECISION
30010 IF N<2 OR N>3 THEN NXT=0:GOTO 3010
0'REM DEATH BY LONELINESS OR OVERCROWDIN
G
30020 IF C=0 AND N=3 THEN NXT=1:GOTO 301
00'REM BIRTH
30030 NXT=C'REM NO CHANGE
30100 RETURN
```

Ispitajmo pažljivije slijed slika (lijevo)

Slika 1 Dvije krajnje ćelije umiru ali na svakoj strani nastaju dvije nove jer svaka ima po tri živa susjeda

Slika 2 Ćelije na uglovima preživljavaju jer imaju samo tri susjeda dok sve ostale umiru od prenapučenosti

Slike 3—6 Oblik prstene se razvije dobro eli sve ćelije unuter njega umiru Ostaju samo četiri grupe od po tri ćelije u redu dvije krajnje umiru od iscrpljenosti ali se istodobno rađaju dvije nove sa svake strane Proces se ponavlja ciklički raspored ćelija naizmjence prelazi iz horizontalnog u vertikalni iz vertikalnog u horizontalni smjer i tako bez prekide pa je stoga takav respored nazvan žmigavcem'

```
x        x      x x      x        x
 x      x x      x        x        x
  x      x       x      x x         x
x x      x      x         x       x x
 0       1      2         3        4
```

Križić — kružić

To je jednostavna igra — ako je znate igrati U tu se igru može uključiti robot (vidi str 140—141) pa umjesto da se na ekranu ispisuju križići ('×') i kružići (o) može se igrati s pravim figurama

```
10 REM ***************************
20 REM NOUGHTS AND CROSSES
30 REM ***************************
40 DIM T(8,3)
50 DIM U(9,4)
60 DIM C(9),B(9)
70 PRINT CHR$(12)
80 PRINT "*** NOUGHTS AND CROSSES ***"
90 PRINT
100 PRINT "YOU ARE 'X', THE COMPUTER IS
'O'"
110 PRINT "BELIEVE IT OR NOT, YOU CAN WI
N!"
120 X1=0:X2=0
130 REM INIT TABLES P AND U
140 FOR P=1 TO 8
150 FOR I=1 TO 3
160 READ T(P,I)
170 NEXT I
180 NEXT P
190 DATA 1,2,3
200 DATA 8,9,4
210 DATA 7,6,5
220 DATA 1,8,7
230 DATA 2,9,6
240 DATA 3,4,5
250 DATA 1,9,5
260 DATA 7,9,3
270 FOR S=1 TO 9
280 FOR J=1 TO 4
290 READ U(S,J)
300 NEXT J
310 NEXT S
320 DATA 1,4,7,0
330 DATA 1,5,0,0
340 DATA 1,6,8,0
350 DATA 2,6,0,0
360 DATA 3,6,7,0
370 DATA 3,5,0,0
380 DATA 3,4,8,0
390 DATA 2,4,0,0
400 DATA 2,5,7,8
410 REM HERE FOR A NEW GAME
420 LET N=0
430 FOR S=1 TO 9
440 LET C(S)=0
450 LET B(S)=0
460 NEXT S
470 REM FLIP COIN FOR WHO STARTS
480 IF RND(1) < .5 THEN 510
490 PRINT "YOU HAVE THE FIRST GO"
500 GOTO 530
510 PRINT "THE COMPUTER HAS FIRST GO"
520 GOTO 760
530 REM PLAYER'S MOVE
540 GOSUB 1230
550 INPUT "YOUR GO";M
560 LET F=-1
570 IF M=INT(M) THEN 600
580 PRINT "ILLEGAL MOVE--TRY AGAIN"
590 GOTO 530
600 IF M<1 OR M>9 THEN 580
610 IF B(M) <> 0 THEN 580
620 REM UPDATE LISTS C, CHECK FOR WIN
630 LET B(M)=F
640 FOR J=1 TO 4
650 LET P=U(M,J)
660 IF P=0 THEN 700
670 LET C(P)=C(P)+F
680 IF C(P)=-3 THEN GOSUB 1230:GOTO 900
690 IF C(P)=3 THEN GOSUB 1230:GOTO 800
700 NEXT J
710 LET N=N+1
720 IF N=9 THEN GOSUB 1230:GOTO 930
730 IF F=1 THEN 530
740 REM COMPUTER'S MOVE
750 GOSUB 1230
760 GOSUB 950
770 PRINT "THE COMPUTER'S MOVE IS:"
780 LET F=1
790 GOTO 620
800 REM GAME OVER
810 PRINT "AND THE COMPUTER WINS"
820 X2=X2+1
830 PRINT
840 PRINT"SCORE NOW--COMPUTER";X2;", YOU
";X1
850 INPUT "TYPE 'YES' FOR ANOTHER GAME";
A$
860 IF A$<>"YES" THEN STOP
870 PRINT
880 PRINT "NEW GAME"
890 GOTO265
900 PRINT "YOU BEAT THE COMPUTER"
910 X1=X1+1
920 GOTO 830
930 PRINT "THE GAME IS A DRAW"
940 GOTO 830
950 REM SELECT A MOVE
960 FOR P=1 TO 8
970 IF C(P)=2 THEN 1030
980 NEXT P
990 FOR P=1 TO 8
1000 IF C(P)=-2 THEN 1030
```

```
1010 NEXT P
1020 GOTO 1070
1030 FOR I=1 TO 3
1040 LET M=T(P,I)
1050 IF B(M)=0 THEN 1220
1060 NEXT I
1070 FOR S=1 TO 9
1080 LET V(S)=0
1090 IF B(S)<>0 THEN 1150
1100 FOR J=1 TO 4
1110 LET P=U(S,J)
1120 IF P=0 THEN 1140
1130 LET V(S)=V(S)+1+ABS(C(P))
1140 NEXT J
1150 NEXT S
1160 LET V=0
1170 FOR S=1 TO 9
1180 IF V(S)<=V THEN 1210
1190 LET V=V(S)
1200 LET M=S
1210 NEXT S
1220 RETURN
1230 REM PRINT BOARD
1240 PRINT
1250 PRINT" 1  2  3"; TAB(15);
1260 FOR A=1 TO 3
1270 GOSUB 1410
1280 NEXT A
1290 PRINT:PRINT
1300 PRINT " 8  9  4";TAB(15);
1310 A=8:GOSUB 1410
1320 A=9:GOSUB 1410
1330 A=4:GOSUB 1410
1340 PRINT:PRINT
1350 PRINT " 7  6  5";TAB(15);
1360 FOR A=7 TO 5 STEP -1
1370 GOSUB 1410
1380 NEXT A
1390 PRINT:PRINT
1400 RETURN
1410 IF B(A)=0 THEN PRINT".  ";:RETURN
1420 IF B(A)=-1 THEN PRINT"X  ";:RETURN
1430 PRINT"O  "; RETURN
```

Zombi

Ovo je vrlo vesela igra: pokušavate se izmaknuti stravičnim bićima koja jure na vas. Ako uspijete, čudovišta propadaju u duboke procijepe i tamo ugibaju. Ako ste nespretni — proždiru vas zadovoljna proteinskim ručkom.

```
10 REM ********************
20 REM      ZOMBIES
30 REM ********************
40 REM 'GRAPH1' TURNS PLOTTING ON
50 REM 'GRAPH0' TURNS IT OFF
60 REM 'PLOT X,Y,Z' PUTS THE CHARACTER W
ITH ASC(I) CODE Z AT X,Y
70 CLEAR 1000
80 PRINT CHR$(12):GOSUB 640
90 GRAPH1
100 CLEAR
110 DIM B(12,22),Z(25,2),P(8),Q(8)
120 FOR N1=1 TO 8
130 READ P(N1),Q(N1)
140 NEXT N1
150 DATA 1,-1,1,0,1,1,0,1,-1,1,-1,0,-1,-
1,0,-1
160 PLOT 74,57,53
170 PLOT 76,57,52
180 PLOT 78,57,51
190 PLOT 74,54,54
200 PLOT 76,54,88
210 PLOT 78,54,50
220 PLOT 74,51,55
230 PLOT 76,51,56
240 PLOT 78,51,49
250 FOR N1=1 TO 25
260 FOR N2=1 TO 2
270 Z(N1,N2)=0
280 NEXT N2
290 NEXT N1
300 Z1=0
310 FOR N1=1 TO 12
320 FOR N2=1 TO 22
330 B(N1,N2)=4
340 PLOT N2*2,N1*3,143
350 NEXT N2
360 NEXT N1
370 FOR N1=2 TO 11
380 FOR N2=2 TO 21
390 R =20*RND(10)
400 IF R>18.5 THEN GOTO 500
410 IF R<17.95 THEN GOTO 480
420 Z1=Z1+1
430 Z(Z1,1)=N1
440 Z(Z1,2)=N2
450 B(N1,N2)=2
460 PLOTN2*2,N1*3,ASC("Z")
470 GOTO 500
480 B(N1,N2)=1
490 PLOTN2*2,N1*3,ASC(" ")
500 NEXT N2
510 NEXT N1
520 X=5+INT(10*RND(15))
530 Y=3+INT(5*RND(15))
540 B(Y,X)=3
550 PLOTX*2,Y*3,ASC("X")
560 FOR N1=Y-1 TO Y+1
570 FOR N2=X-1 TO X+1
580 IF ABS(Y-N1)+ABS(X-N2)=0 THEN GOTO 6
10
590 B(N1,N2)=1
600 PLOTN2*2,N1*3,ASC(" ")
610 NEXT N2
620 NEXT N1
630 GOTO 770
640 PRINT"You have just landed on ZOMBIE
 ISLAND"
650 PRINT
660 PRINT"  Your only hope of survival i
s to lure"
670 PRINT"all the ZOMBIES into potholes
 You may"
680 PRINT"indicate the direction of your
 move"
690 PRINT"as follows."
700 PRINT
710 PRINT"543"
720 PRINT"6X2"
730 PRINT"781":PRINT:PRINT
740 INPUT"Hit RETURN to continue";AA$
750 PRINT CHR$(12)
760 RETURN
770 REM
780 FOR N1=1 TO Z1
790 IFB(Z(N1,1),Z(N1,2))=2 THEN GOTO860
800 FOR N2=N1 TO Z1
810 Z(N2,1)=Z(N2+1,1)
820 Z(N2,2)=Z(N2+1,2)
830 NEXT N2
840 Z1=Z1-1
850 GOTO 780
860 NEXT N1
870 PRINT
880 PRINT"  Your move";
890 INPUT A
900 IF A>8 THEN GOTO 920
910 IF A>=1 THEN GOTO 940
920 PRINT"Please use a number between 1
and 8"
930 GOTO 880
940 B(Y,X)=1
950 PLOTX*2,Y*3,ASC(" ")
960 Y=Y+Q(A)
970 X=X+P(A)
980 ON B(Y,X) GOTO 990,1020,990,1040
```

```
990 B(Y,X)=3
1000 PLOTX*2,Y*3,ASC("X")
1010 GOTO 1060
1020 PRINT"MMMMM TASTY!--Munch Munch--PR
OTOPLASM!"
1030 GOTO 1430
1040 PRINT"aaaaargh HASPLUTCH Right into
 the pit---";
1050 GOTO 1430
1060 Z2=1
1070 Z8=Z(Z2,1)
1080 Z9=Z(Z2,2)
1090 B(Z8,Z9)=1
1100 PLOT Z9*2,Z8*3,ASC(" ")
1110 Z8=Z8+SGN(Y-Z8)
1120 Z9=Z9+SGN(X-Z9)
1130 ON B(Z8,Z9) GOTO 1320,1280,1210,114
0
1140 PRINT"KERSPLOSH---Goes a ZOMBIE"
1150 FOR Z3=Z2 TO Z1
1160 Z(Z3,1)=Z(Z3+1,1)
1170 Z(Z3,2)=Z(Z3+1,2)
1180 NEXT Z3
1190 Z1=Z1-1
1200 GOTO 1370
1210 PLOT X*2,Y*3,42
1220 FOR MN=0 TO 150
1230 NEXT MN
1240 PLOT X*2,Y*3,32
1250 PLOT X*2,Y*3,ASC("Z")
1260 PRINT"The ZOMBIES get lunch after a
ll!"
1270 GOTO 1430
1280 PRINT"EEK!--Here come the ZOMBIES!!
";
1290 B(Z(Z2,1),Z(Z2,2))=2
1300 PLOTZ(Z2,2)*2,Z(Z2,1)*3,ASC("Z")
1310 GOTO 1360
1320 B(Z8,Z9)=2
1330 PLOT Z9*2,Z8*3,ASC("Z")
1340 Z(Z2,1)=Z8
1350 Z(Z2,2)=Z9
1360 Z2=Z2+1
1370 IF Z2<=Z1 THEN GOTO 1070
1380 IF Z1>=1 THEN GOTO 870
1390 PRINT
1400 PRINT"Well done!---You escaped---Th
e"
1410 PRINT"ZOMBIES are extinct."
1420 GOSUB 1550
1430 PRINT" Another game ";
1440 REM
1450 INPUT A$
1460 IF A$="Y" OR A$="YES" THEN PRINT CH
R$(12)  GOTO 30
1470 IF A$="N" OR A$="NO" THEN 1520
1480 PRINT"Please answer YES or NO":GOTO
 1430
1490 GRAPH0
1500 RUN
1510 GOTO 1430
1520 GRAPH0
1530 PRINT CHR$(12)
1540 END
1550 FOR J=X+1 TO 79
1560 FOR HJ=0 TO 1
1570 PLOT J*2,Y*3,ASC("X")
1580 PLOT (J-1)*2,Y*3,ASC(" ")
1590 IF J*2>=78 THEN 1620
1600 NEXT HJ
1610 NEXT J
1620 PLOT J*2,Y*3,ASC(" ")
1630 RETURN
```

Nekoliko programa za ZX Spectrum

Prema neslužbenim procjenama kod nas je oko 90% svih kućnih računala Sinclairove proizvodnje, pa je redakcija hrvatskog izdanja odlučila da čitateljima stavi na raspolaganje nekoliko programa posebno prilagođenih računalu Sinclair ZX Spectrum

Život

```
10 REM ZIVOT
20 DIM S(29,29) DIM ...
40 LET Y=0 LET A=0 LET ...
50 FOR 09=1 TO 29
60 FOR 08=1 TO 29
70 LET S(08,09)=0
80 NEXT 08
90 NEXT 09
100 INPUT "KOJU GENERACIJU PRVU DA
 ISPISEM ";T
120 INPUT "ZADNJU ";R1
140 INPUT "A KOLIKO DA IH PRESKOCIM ";D
160 LET D=D+1
170 PRINT "UNESITE U CRAM * + ILI
 ZAVRSITE SA GOTOVO"
180 FOR M=1 TO 28
190 INPUT M;M$
200 IF M$="GOTOVO" THEN GOTO 320
210 IF LEN M$ 29 THEN GOTO 240
220 PRINT "MAKSIMUM ... ELIJA U LINIJI
 MOLIM PONOVITE"
230 GOTO 190
240 FOR R=1 TO LEN M$
250 IF M$(R TO R)=" " THEN GOTO 300
260 IF M$(R TO R)<>"*" THEN GOTO 3..
270 LET S(M,R)=1
280 LET A=A+1
290 IF R>S THEN LET S=R
300 NEXT R
310 NEXT M
320 GOTO 370
330 PRINT M$(R TO R) ... NIJE NITI ...
 NITI ... PONOVITE LINIJU "
350 GOTO 190
370 IF NOT Y THEN GOTO 390
380 GOTO 20
390 LET Y=0 LET G=0
400 LET Y=M-1 LET N1=M-1
410 PRINT "GENERACIJA = 0 POPULACIJA
 =" A
420 GOSUB 890
430 LET G=G+1
440 LET A=0 LET Y=0
450 FOR 09=1 TO 29
460 FOR 08=1 TO 29
470 LET S(08,09)=0
480 NEXT 08
490 NEXT 09
500 FOR M=1 TO N1
510 FOR R=1 TO S
520 LET J=0
530 IF M 2A OR R 2 THEN GOTO 12..
540 IF NOT X(M,R+1) OR R= ... THEN GOTO
 560
550 LET J=J+1
560 IF NOT X(M+1,R+1) OR M=N1 OR R= ...
 THEN GOTO 580
570 LET J=J+1
580 IF NOT X(M+1,R) OR M=N1 THEN ...
 600
590 LET J=J+1
600 IF M=N1 OR R=1 THEN GOTO 630
610 IF NOT X(M+1,R-1) THEN GOTO 63.
620 LET J=J+1
630 IF R=1 THEN GOTO 660
640 IF NOT X(M,R-1) THEN GOTO 6..
650 LET J=J+1
660 IF M=1 OR R=1 THEN GOTO 680
670 IF NOT X(M-1,R-1) THEN GOTO 69.
680 LET J=J+1
690 IF M=1 THEN GOTO 7..
700 IF NOT X(M-1,R) THEN GOTO 72.
710 LET J=J+1
720 IF M=1 OR R=S THEN GOTO 750
730 IF NOT X(M-1,R+1) THEN GOTO ...
740 LET J=J+1
750 IF J=3 THEN GOTO 7..
760 IF J<>2 OR NOT X(M,R) THEN GOTO ...
770 LET S(M,R)=1
780 LET A=A+1
790 NEXT R
800 NEXT M
810 IF A=0 THEN GOTO 12..
820 IF G=R1 THEN GOTO 84.
830 IF G<T OR (G-1)/D<>INT ((G-1)/D)
 THEN GOTO 860
840 PRINT "GENERACIJA = " G
 "POPULACIJA = " A
```

```
850 GOTO 870
860 LET V=9
870 GOSUB 890
880 GOTO 430
890 LET J=50 LET H=50
900 LET C=0:LET D=0
910 FOR M=1 TO N1
920 FOR R=1 TO S
930 IF  NOT S[M,P] THEN GOTO 1000
940 IF M<J THEN LET J=M
950 IF P<H THEN LET H=P
960 IF M>C THEN LET C=M
970 IF R>D THEN D=R
980 IF V=9 THEN GOTO 1000
990 PRINT TAB 2*P-2;"*";
1000 NEXT P
1010 IF V=9 THEN GOTO 1030
1020 PRINT
1030 NEXT M
1040 IF G=R1 THEN GOTO 1270
1050 LET C=C-J+3
1060 LET D=D-H+3
1070 FOR 09=1 TO 29
1080 FOR 08=1 TO 29
1090 LET Y[08 09]=0
1100 NEXT 08
1110 NEXT 09
1120 FOR M=1 TO N1
1130 FOR R=1 TO S
1140 IF  NOT S[M P] THEN GOTO 1160
1150 LET X[M J+2 P-H+2]=1
1160 NEXT R
1170 NEXT M
1180 FOR 09=1 TO 29
1190 FOR 08=1 TO 29
1200 LET S[08 09]=0
1210 NEXT 08
1220 NEXT 09
1230 LET N1=C
1240 LET S=D
1250 RETURN
1260 PRINT  NEMA VISE ZIVOTA I
 GENERACIJI   G
1270 PRINT "**GOTOVO**"
1280 STOP
1290 PRINT  SKUPINE SU PRESLE LIMIT
 VOG PROGRAMA
1300 PRINT  MORAMO ZAVRSITI
```

Rođendani

```
100 REM RODJENDANI
110 PRINT "PROGRAM VRSI DVA PRORACUNA
120 PRINT "1-  IZRACUNAVA KOLIKO LJUDI
 MORA BITI PRISUTNO
130 PRINT "    U JEDNOJ GRUPI DA
 POSTOJI  X% VJEROJATNOSTI DA"
140 PRINT "    IMAJU RODJENDAN ISTOG
 DATUMA"
150 PRINT  2   IZRACUNAVA VJEROJATNOST
 UD X% LJUDI PRISUTNO
160 PRINT "    U JEDNOJ GRUPI DA IMAJU
 RODJENDAN ISTOG DATUMA   PRINT
170 INPUT  KOJI PRORACUN ZELITE  1 ILI
 2)   G0
190 IF G0=1 THEN GOTO 210
200 IF G0=2 THEN GOTO 310
205 GOTO 170
210 INPUT  ZADAJTE POSTOTAK VJEROJATNOS
 TI   Z9
220 FOR S1=1 TO 400  LET S2=366-S1
 LET S3=1
230 FOR S4=365 TO S2 STEP -1  LET
 S3=S3*S4/365  NEXT S4
240 LET X=INT((INT(1000*(1-S3))/10)+1
 0)
250 IF X=29 THEN GOTO 290
260 IF X 29 THEN GOTO 280
270 PRINT "MORA BITI OTPRILIKE
 ";S1-1;" LJUDI PRISUTNO"  GO TO 300
280 NEXT S1
290 PRINT  MORA BITI   S1   LJUDI
 PRISUTNO
300 PRINT "DA POSTOJI VJEROJATNOST OD
 " Z9 "%" "  STOP
310 INPUT "KOLIKO IMA LJUDI U GRUPI" S1
320 LET S2=366-S1  LET S3=1
330 FOR S4=365 TO S2 STEP -1  LET
 S3=S3*S4/365  NEXT S4
340 PRINT "VJEROJATNOST JE " ;INT(1000*
 (1-S3))/1000)*100  %
```

Križić — kružić

```
10 REM TRODIMENZIONALNI KRIZIC-KRUZIC
20 DIM Q(16) DIM R(76 4) DIM S(64) DIM
 T(76) DIM U(5 3)
30 DEF FNS(Z)=S[R[Z 1]]+S[R[Z
 2]]+S[R[Z 3]]+S[R[Z 4]]
40 DEF FNM(Z)=Z+11(0+6*INT(Z/4- 2 +6*IN
 T(Z 16- 05)
50 DEF FNU(Z)=INT T[Z]-5*INT T[Z]/5)
60 FOR 01=1 TO 16
70 READ 00
80 LET Q[01]=00
90 NEXT 01
100 FOR 01=1 TO 76
110 FOR 02=1 TO 4
120 READ 00
130 LET R[01 02]=00
140 NEXT 02
150 NEXT 01
160 FOR 01=1 TO 5
170 FOR 02=1 TO 3
180 LET U[01 02]=0
190 NEXT 02
200 NEXT 01
210 LET E=0 LET G3=0
220 INPUT  VASE IME   B$
240 PRINT "OVO JE IGRA KRIZIC-KRUZIC
 ";B$ "
250 INPUT "ZELITE LI UPUTE ?  A$
270 IF A$ <>"DA" THEN GOTO 290
280 PRINT  OVO JE KLASICNA IGRA
 KRIZIC-KRUZIC ALI
281 PRINT "TRODIMENZIONALNA  NA 4<4<4
 POLJA  KOCKA) "
282 PRINT "OSVAJAJU SE SVE DIMENZIJE
 PROSTORNE I PLOSNE
283 PRINT  ISPISUJU SE 4 PLOCE 4<4
284 PRINT "PRILIKOM UNOSENJA POTEZA "
285 PRINT "UNOSITE BROJ OD TRI
 ZNAMENKE GDJE JE
286 PRINT "PRVA ZNAMENKA = BROJ PLOCE"
287 PRINT  DRUGA = RED NA PLOCI"
288 PRINT "TRECA = KOLONA NA PLOCI
289 PRINT "TKO PRVI POVUCE DIJAGONALU
 OD 4 POZICIJE POBJEDNIK JE "
290 INPUT  KADA STE VI NA POTEZU DA
 LI ISPISEM TABLU ?" C$
310 IF C$ <>"DA" THEN GOTO 330
320 PRINT "VASE POZICIJE SU OZNACENE
 SA  X     A MOJE SA  O
330 PRINT
340 PRINT  "IZABERITE STUPANJ STRATEGIJE
 IZMEDJU 1 I 5  "
350 PRINT " 1 JE NAJLAKSA   5 JE
 NAJTEZA"
370 PRINT  KOJI STUPANJ      B$
380 INPUT L
390 IF ABS L   INT L-1  AND L < THEN
 GOTO 420
400 PRINT   IZMEDJU 1 I 5 MOLIM "
410 GOTO 370
420 LET Q=2 LET U0=2
430 LET A3=16
440 LET P=INT  6*RND +1
450 FOR U1=1 TO 64
460 LET S[U1]=0
470 NEXT U1
480 FOR 01=1 TO 76
490 LET T[01]=0
500 NEXT 01
510 PRINT "STUPANJ  L    ZELITE LI
 PRVI POVUCI POTEZ "
520 INPUT A$
530 IF A$="DA"  THEN GOTO 550
540 IF INT(L 2+ 5)=1 THEN GOTO 2110
541 IF INT(L/2+ 5)=2 THEN GOTO 2580
542 IF INT(L 2+ 5)=3 THEN GOTO 1510
550 LET D=1
560 GOTO 590
570 IF L$<>"DA  THEN GOTO 590
580 GOSUB 2820
590 PRINT "VAS POTEZ  " B$
600 LET E=E+RND
610 INPUT F
620 LET E1=INT(F/100)
630 LET E2=INT(F/10)-10*E1
640 LET E3=F-100*E1-10*E2
650 IF E1<5 AND E2 5 AND E3 5 AND
 ABS(F-111)=INT(F-111)  THEN GOTO 680
660 PRINT  NEPRAVILAN POTEZ  PONOVITE
 MOLIM "
670 GOTO 590
680 LET M=16*E1+4*E2+E3-20
690 IF S[M]<1 THEN GOTO 720
700 PRINT "TA POZICIJA JE ZAUZETA
 PONOVITE "
710 GOTO 590
720 LET S[M]=1
730 LET Q=Q+2
740 GOSUB 2200
750 GOSUB 2100
760 IF Q=3 AND (L=3 OR L=4  THEN GOTO
 2580
```

```
770 LET A=0 LET B=0
780 FOR F=1 TO 76
790 IF T[F] 4 THEN
800 PRINT "CESTITAM,
810 LET G3=G3+1
820 LET U[L,3]=U[L,3]+
830 PRINT "POBIJEDILI
 ) FNM(R[F 2]) FNM R(
840 PRINT FNM(R[F 4])
 POZICIJA :
 GOSUB 2(20
 FOR A= TO 5
 FOR B=1 TO 3
 IF U[A B] 0 THEN
 NEXT B
 GOTO 950
 PRINT "[AKLE, NA ST
 PRINT "VI STE POBIJE
 PUTA"
 PRINT "JA SAM POBI
 " PUTA"
 IF [A,2]=0 THEN GOT
 PRINT "REMIZIRALI
940 PRINT
951 NEXT A
 PRINT B$ ZELITE
 IGRU ?"
 INPUT A$
 IF A$= [A THEN
 STOP
 IF T[F] 15 THEN
 LET A=F
 IF T[F] THEN
1030 LET B=F
1040 NEXT F
1050 IF A=0 THEN GOTO 115
1060 GOSUB 1990
1070 IF D0=2 THEN GOTO 1100
1080 PRINT "ZAO MI JE B$ "
1090 PRINT "BAS KAO PEKON ";
1100 PRINT "POVUKAO SAM " FNM R(
 ) " I";
1110 LET F=A
1120 LET U[L,1]=U[L,1]+1
1130 LET S[R[A,F]]=5
1140 GOTO 830
1150 IF D<65 THEN GOTO 124
1160 IF A=66 THEN GOTO 1210
1170 FOR F=1 TO 64
1180 IF S[F]=0 THEN GOTO 120
1190 NEXT F
1200 PRINT "VUCEM NA ";FNM F)
1210 PRINT "REMI "
1220 LET U[L,2]=U[L,2]+1
1230 GOTO 860
1240 IF L=1 THEN GOTO 2400
1250 IF B=0 THEN GOTO 1310
1260 LET A=B
1270 GOSUB 1990
1280 PRINT "LIJEPI POKUSAJ B$
 --JA VUCEM NA " FNM(R[A C])
1290 LET S[R[A 1]]=5
1300 GOTO 570
1310 IF L=2 THEN GOTO 2400
1320 FOR Z=5 TO 1 STEP -4
1330 FOR F=1 TO 76
1340 LET Y=FNS(F)
1350 IF INT(Y) 2*Y THEN GOTO 1410
1360 IF Y 2*Z THEN GOTO 1760
1370 FOR G=1 TO 4
1380 IF S[R[F G]]>0 THEN GOTO 1400
1390 LET S[R[F,G]]= 1
1400 NEXT G
1410 NEXT F
1420 GOSUB 2160
1430 FOR F=1 TO 76
1440 IF T[F]- 4 OR T[F]=Z+ 3 THEN GOT
 1900
1450 NEXT F
1460 IF Y=1 THEN GOTO 1490
1470 IF L=3 THEN GOTO 2570
1480 GOSUB 2200
1490 NEXT Y
1500 IF L=4 THEN GOTO 2570
1510 FOR N=1 TO 69 STEP 4
1520 LET P=INT(T[N])+INT(T[N+1])+INT(T[
 N+2])+INT(T[N+3])
1530 IF P<>4 AND P 9 THEN GOTO 1600
1540 LET J- 1
1550 FOR F=N TO N+3
1560 GOTO 1910
1570 NEXT F
1580 LET J=0
1590 GOTO 1550
1600 NEXT N
1610 FOR F=1 TO 16
1620 IF S[Q[F]]> 5 THEN GOTO 1660
1630 LET S[Q[F]]=5
1640 PRINT "VUCEM NA ";FNM(Q[F])
1650 GOTO
1660 NEXT F
1670 LET S=0
1680 FOR F=1 TO INT( *RND +
1690 LET S=S+1
1700 IF S[S] 5 THEN GOTO 1720
1710 GOTO 1690
1720 NEXT F
1730 PRINT "SVIDJA MI SE ";FNM(S
1740 LET S[S]=5
1750 GOTO 570
1760 FOR G=1 TO 4
1770 IF S[R[F G]]<> 1 THEN GOTO
1780 LET S[R[F,G]]=5
1790 IF Y<5 THEN GOTO 1860
 IF G3=1 THEN GOTO
1810 PRINT B$ VEC STE NE POB
 ALI VISE NE VIDI
1820 LET Q0=1
1830 GOTO 1870
1840 PRINT "OH VIDINO
 IZVUCI OD OPOG
1850 GOTO 1870
1860 PRINT "MUCKICE! U
 TRENUTKU
1870 PRINT " VUCEM NA " FNM R(
 GOTO 571
 NEXT G
 LET J= 1
 LET K=INT( INT F 4 14
 FOR G= TO 5 STEP 5
 IF S[R[F G]] 0 THEN GOTO
1940 PRINT " UZIMAN ";FNM(R[F G])
1950 LET S[R[F ]]=5
1960 GOTO 570
1970 NEXT G
 GOTO 1570
 FOR C=1 TO 4
2000 IF S[R[A ]] 5 THEN
 NEXT C
 LET A$=
 PRINT
 FOR F=1 TO 13 STEP 4
 FOR G=F TO F+48 STEP 16
 FOR N=G TO G+3
 LET N=ABS(INT(S[N])-2)
 PRINT A$[N TO N]
 NEXT N
2100 PRINT
2110 NEXT G
2120 PRINT
2130 NEXT F
 140 PRINT
2150 RETURN
2160 FOR W=1 TO 76
2170 LET T[W]=FNS(W)
2180 NEXT W
2190 RETURN
2200 FOR W=1 TO 64
2210 LET S[W]=INT(S[W])
 NEXT W
2230 RETURN
2240 DATA 1 49 52 4 13,61 64 1
 3 38,26 42 7 43 1 2 3 4 5 6
2250 DATA 8 9 10,11 12 13 14 15
 18 19 20 21 22 23 24 25
2260 DATA 29 30 31 32 33 34 35
 39 40 41,42 43 44 45 46
2270 DATA 49 50 51 52 53 54
 58 59 60 61 62 63 64,1 17 3
2280 DATA 18 34,50 3,19 35 51 4
 52 5 21 37,5 6 22 38 54
2290 DATA 55 8 24 40 56 9 25 ,1
 42 58 11 43 59 12 28 44
2300 DATA 13 29 45 61 14 30 46
 31,47 6 16 48 64 1 5 9 1
2310 DATA 21 25 29 33 37 41 45
 57 61 2 6 10 14,18 22 26 30 34
2320 DATA 42 46 50 54 58 62 3 7
 19 23 27 31 35 39 43 47 51 55
2330 DATA 63 4 8 12 16 20 24 28
 40,44 48 52 56 60 64 1 6 11 16
2340 DATA 22 27 32 33 38 43 48
 59 64 13 10 7 4 29 26 23,20 45
2350 DATA 39 36 61 58 55 52 1 2
 2,22,42 62 3,23,43 63,4 24 44
2360 DATA 49 37 25 13 50 38 26
 39 27,15 52 40 28 16 1 19 35 52
2370 DATA 22 39 56 9 26 43 60 13
 64 49,34 19 4 52 38,23 8 57 42
2380 DATA 27 12 61 46 31 16 1
 16 27 38 49 4 23 42 61 13 26
2390 DATA 52
2400 IF R=0 THEN GOTO 1670
2410 IF FNG(P) THEN GOTO 2460
2420 LET A=R
2430 GOSUB 1990
2440 PRINT "IDEM NA " FNM R[A C]
2450 GOTO 1290
```

```
2460 LET A=0 LET  =0
2470 FOR F=1 TO 76
2480 IF FNG(F) THEN GOTO 2500
2490 LET A=A+1
2500 NEXT F
2510 IF A=0 THEN GOTO 1670
2520 FOR F=1 TO INT(A*RND)+1
2530 LET R=R+1
2540 IF FNG(R) THEN GOTO 2530
2550 NEXT F
2560 GOTO 2420
2570 IF FNG(R7)=0 AND FNG R8)=0 THEN
 GOTO 2660
2580 LET P9=R9+1
2590 IF R9>5775 THEN GOTO 1670
2600 LET R7=INT(R9/76- 01 +1
2610 LET R8=R9-76*R7+76
2620 IF FNG(R8) OR (INT(R  4   2 -INT R8
 4- 2)) THEN GOTO 2590
2630 IF  NOT FNG(R7) THEN GOTO 2600
2640 LET R9=R9+77 P9
2650 GOTO 2590
2660 FOR F=1 TO 4
2670 FOR G=1 TO 4
2680 IF R[R7 F]-R[R8 G] THEN GOTO 2 
2690 NEXT G
2700 NEXT F
2710 GOTO 2580
2720 IF S[R[P  F]]  5 THEN GOTO 
2730 LET R=R7
2740 IF T[R] 10 THEN GOTO 2770
2750 LET R=R8
2760 LET F=G
2770 FOR H=1 TO 4
2780 IF S[R[P H]]> 5 OR H=F THEN GOTO
 2820
2790 PRINT "ODLAZIM NA   FIN PLP H
2800 LET S[R[R,H]]=5
2810 GOTO 570
2820 NEXT H
```

Okretanje

```
100 REM *** OKRETANJE
130 DIM A[20]
160 INPUT "ZELITE LI UPUTE ?" A$
170 IF A$="DA" THEN GOSUB 710
180 INPUT "SA KOLIKO BROJEVA ZELITE
 POKUSATI (1-20)?" N
190 IF N<0 OR N>20 THEN GOTO 180
210 FOR I=1 TO N
215 LET A[I]=I
220 NEXT I
230 FOR I=N TO 2 STEP -1
235 LET K=INT(I*RND)+1
240 LET T=A[I]
245 LET A[I]=A[K]
250 LET A[K]=T
255 NEXT I
300 PRINT "POCINJEMO      EVO BROJEVA
310 LET T=0
320 GOSUB 610
330 INPUT "KOLIKO DA OKRENEM ?" R
350 IF R =0 THEN GOTO 9999
360 IF R  = N THEN GOTO 390
370 PRINT "OPALA ! IMATE SAMO   N
 BROJEVA"
380 GOTO 330
390 LET T=T+1
410 FOR K=1 TO INT(R/2)
420 LET Z=A[K]
430 LET A[K]=A[R-K+1]
440 LET A[R-K+1]=Z
450 NEXT K
460 GOSUB 610
480 FOR K=1 TO N
490 IF A[K] K THEN GOTO 770
500 NEXT K
510 PRINT "USPJELI STE U   T
 POKUSAJA !"
520 STOP
610 PRINT
620 FOR K=1 TO N
630 PRINT A[K];
640 NEXT K
650 PRINT
670 RETURN
710 PRINT
720 PRINT "OVA SE IGRA ZOVE  REVERSE
 (OKRETANJE)"
730 PRINT "VAS ZADATAK JE DA U STO
 MANJEM BROJEM POKUSAJA SREDITE
740 PRINT "LISTU BROJEVA  PO VELICINI
 SLIJEVA NA DESNO
750 PRINT "UHUSITE KOLIKO CE SE PRVIH
 BROJEVA OKRENUTI"
760 PRINT  (PRVIH S LIJEVE STRANE)
770 PRINT  NA PRIMJER  "
780 PRINT " 2 3 4 5 1 6 7 8 9
790 PRINT
800 PRINT  AKO UTIPKATE 4  REZULTAT
 JE:
810 PRINT
820 PRINT    5 4 3 2 1 6 7 8 9
830 PRINT
840 PRINT  I SADA AKO OTIPKATE 5
 USPJELI STE !
850 PRINT
860 PRINT    1 2 3 4 5 6 7 8 9
870 PRINT
890 PRINT "AKO ZELITE ZAVRSITI
 OTIPKAJTE 0"
910 RETURN
9999 END
```

Rat

```
10 REM AMERICKI  GRADJANSKI RAT
160 LET Z=0 LET P2=0 LET T1=0 LET
 T2=0 LET Q1=0  LET M3=0
165 LET M4=0 LET L=0 LET W=0 LET
 P1=0 LET P1=0
170 INPUT  ZELITE LI KOMENTARE
 POJEDINIH BITAKA   Z$
190 IF Z$="DA" THEN LET Z=1
210 PRINT
240 RESTORE
250 IF Z=1 THEN GOTO 500
260 PRINT  OVO JE SIMULACIJA AMERICKOG
 GRADJANSKOG RATA "
270 PRINT "DA BISTE IGRALI  POTREBNO
 JE SAMO DA ODGOVARATE NA PITANJA
280 PRINT "UPAMTITE DA SU SVI FAKTORI
 MEDJUSOBNO POVEZANI I DA ZASI
290 PRINT "ODGOVORI MOGU IZMIJENITI
 POVIJESNA ZBIVANJA DOGADJAJI !
300 PRINT "UVJETI AUTENTICNI SU VECINA
 BITAKA TEZI ISHOD "
310 PRINT "POVIJESNOM ISHODU ALI
 KONACNI REZULTAT IPAK OVISI O VAMA
320 PRINT
330 PRINT "SVRHA JE IGRE POBIJEDITI U
 STO VISE BITAKA
340 PRINT
350 GOTO 490
360 PRINT  VASE MOGUCNOSTI ZA
 DEFANZIVNU STRATEGIJU SU SLIJEDECE
370 PRINT               (1) ARTILJERIJSKI
 NAPAD
380 PRINT "             (2) UTVRDJIVANJE
 PROTIV FRONTALNOG NAPADA
390 PRINT "             (3) UTVRDJIVANJE
 PROTIV BOCNOG NAPADA"
400 PRINT "             (4) POVLACENJE"
410 GOTO 470
420 PRINT "VASE MOGUCNOSTI ZA
 OFANZIVNU STRATEGIJU SU SLIJEDECE
430 PRINT           (1) ARTILJERIJSKI
 NAPAD
440 PRINT "           2  FRONTALNI NAPAD
450 PRINT "           3  NAPAD  BOKA
460 PRINT          4  OPKOLJAVANJE
470 PRINT  MOZETE SE PREDATI AKO
 UTIPKATE  5  ZA VASU STRATEGIJU
480 RETURN
490 PRINT  VI STE KONFEDERACIJA
 GREINO!"
500 READ M1 M2 C1 C2 M A U
510 LET I1=10+(L-W)*2
520 LET I2=10+(W-L)*2
530 LET D1=100*INT((M1*(100-I1   20)/
 (1+(R1-Q1)/(R1+1))+ 5)
540 LET D2=100*INT(M2*(100 I2  20))+ 5
550 LET F1=5*M1/6
560 LET A1=Z
590 PRINT "NALAZITE SE PRED BITKOM
600 GOSUB 1660
610 PRINT "        KONFEDERACIJA
 UNIONISTI"
620 PRINT "LJUDI " INT(M1*(1+(P1 T1    M
 3+1)))
630 PRINT INT(M2*(1+(P2-T2)/(M4+1)))
640 PRINT "NOVAC $" D1    $  D2
650 PRINT "INFLACIJA   I1+ 5  %  "
 " I2 "
660 PRINT
670 INPUT "KOLIKO ZELITE POTROSITI ZA
 HRANU  ,F
690 IF F<0 THEN GOTO 1630
700 INPUT "KOLIKO CETE DATI ZA PLACE  S
720 IF S 0 THEN GOTO 1630
730 INPUT "I NA KRAJU  KOLIKO DAJETE
 ZA MUNICIJU " B
750 IF B<0 THEN GOTO 1630
760 PRINT
770 IF F+S+B  = D1 THEN GOTO 810
```

```
780 PRINT "RAZMISLITE
UKUPNO IMATE SAMO $";
790 GOTO 660
800 LET O=((2*F^2+S^2)/F1^2+1)
810 IF O<10 THEN GOTO 840
820 PRINT "MORAL JE VISOK"
830 GOTO 880
840 IF O<5 THEN GOTO 870
850 PRINT "MORAL JE ZADOVOLJAVAJUCI"
860 GOTO 880
870 PRINT "MORAL JE NIZAK "
880 IF M<>3 THEN GOTO 940
890 PRINT "VI STE U OFENZIVI"
900 PRINT
910 GOSUB 420
920 PRINT
930 GOTO 1030
940 IF M<>1 THEN GOTO 1000
950 PRINT "VI STE U DEFENZIVI"
960 PRINT
970 GOSUB 360
980 PRINT
990 GOTO 1030
1000 PRINT "OBJE STRANE SU U OFENZIVI"
1010 PRINT
1020 GOSUB 420
1030 PRINT
1040 INPUT "VASA STRATEGIJA ";Y
1060 IF Y=5 THEN GOTO 2640
1070 IF ABS(Y-3)<3 THEN GOTO 1100
1080 PRINT "HAZARDERU!! UPOTRIJEBI
PAK DRUGU STRATEGIJU !!"
1090 GOTO 1030
1100 PRINT
1110 PRINT "          KONFEDERALIJA
UNIONISTI"
1120 LET C5=(2*C1/5)*(1+1/(2*(ABS(INT(4*
RND(1)+1)-Y)+1)))
1130 LET C5=INT(C5*(1+1/O)*(1.28+F1/(B+1
)))+.5)
1140 IF C5+100/O<M1*(1+(P1-T1)/(M3+1)) T
HEN GOTO 1190
1150 LET C5=INT(13*M1/20*(1+(P1-T1)/(M3+
1)))
1160 LET E=7*C5/13
1170 LET U=1
1180 GOTO 1200
1190 LET E=100/O
1200 PRINT "GUBICI", C5 [illegible]
*20)+.5)
1210 PRINT "DEZERTERA", INT(E) [illegible]
1220 PRINT
1230 IF C5-C1 = 0 THEN GOTO [illegible]
1240 PRINT "VASI GUBICI SU" [illegible]
-C5)/C1+.5) "% MANJI NEGO"
1250 GOTO 1270
1260 PRINT "VASI GUBICI SU" [illegible]
C1)/C1+.5) "% VECI NEGO"
1270 PRINT "STVARNI GUBICI U BITCI"
1280 LET A1=1
1290 GOSUB 1660
1300 IF U=1 THEN GOTO 1320
1310 IF C5+E<17*C2*C1/(C5*20)+5+O THEN
GOTO 1350
1320 PRINT "IZGUBILI STE BITKU"
1330 LET L=L+1
1340 GOTO 1370
1350 PRINT "POBIJEDILI STE U BITCI"
1360 LET W=W+1
1370 GOSUB 1660
1380 IF W=8 THEN GOTO 2650
1390 LET T1=T1+C5+E
1400 LET T2=T2+17*C2*C1/(C5*20)+5+O
1410 LET P1=P1+C1
1420 LET P2=P2+C2
1430 LET Q1=Q1+(F+S+B)
1440 LET R1=R1+M1*(100-I)/20
1450 LET M3=M3+M1
1460 LET M4=M4+M2
1470 IF A=14 THEN GOTO 2670
1480 GOTO 500
1490 DATA 18000,18500,1967,2708,1,0
1500 DATA 40000,44094,10699,13047,3
2,0
1510 DATA 95000,115000,20614,15849,3
3,0
1520 DATA 54000,63000,10000,14000,2
4,0
1530 DATA 40000,50000,10000,12000,3
5,0
1540 DATA 75000,120000,5377,12653,1
6,0
1550 DATA 38000,45000,11000,12000,1
7,0
1560 DATA 32000,90000,13000,17197,2,8,0
1570 DATA 50000,70000,12000,19000,1
9,0
1580 DATA 72500,85000,20000,23000,3
10,0
[illegible]
1690 DATA [illegible]
1700 DATA [illegible]
13,0
1710 DATA [illegible]
14,0
1720 [illegible]
1730 GOTO 2620
1740 IF A<>2 THEN GOTO 1810
1750 PRINT "KOD SHILOH-A"
1760 PRINT
1770 IF A1=1 THEN GOTO 2620
1780 PRINT "6-7 TRAVNJA 1862
IZNENADNI NAPAD KONFEDERALIJE"
1790 PRINT "NA SHILOH NEUSPIO ZBOG
SLABE ORGANIZACIJE"
1800 GOTO 2620
1810 IF A<>3 THEN GOTO 1890
1820 PRINT "OD SEDAM DANA"
1830 PRINT
1840 IF A1=1 THEN GOTO 2620
1850 PRINT "25 LIPANJ-1 SRPANJ 1862
GENERAL LEE PODRZAVA"
1860 PRINT "SNAGE GENERALA MCCLELLAN
I U ZESTOKOM NAPADU"
1870 PRINT "IZBACUJE UNIONISTE IZ
RICHMOND-A"
1880 GOTO 2620
1890 IF A<>4 THEN GOTO 1970
1900 PRINT "DRUGI BULL RUN"
1910 PRINT
1920 IF A1=1 THEN GOTO 2620
1930 PRINT "29-30 KOLOVOZ 1862
UJEDINJENE SNAGE KONFEDERALIJE"
1940 PRINT "POD LEE EM I JACKSON
POTISKUJU UNIONISTE NATRAG "
1950 PRINT "U WASHINGTON"
1960 GOTO 2620
1970 IF A<>5 THEN GOTO 2030
1980 PRINT "KOD ANTIETAM-A"
1990 IF A1=1 THEN GOTO 2620
2000 PRINT "17 LISTOPAD 1862
UNIJACI POKUSAVAJU PRIPOJITI"
2010 PRINT "MARYLAND KONFEDERALI I"
2020 GOTO 2620
2030 IF A<>6 THEN GOTO 2120
2040 PRINT "KOD FREDERICKSBURG-A"
2050 IF A1=1 THEN GOTO 2620
2060 PRINT "13 PROSINAC 1862"
2070 PRINT "KONFEDERALIJA POD KOMANDOM
GENERALA LEE-A USPIJEVA "
2080 PRINT "ODBIJA NAPAD UNIONISTA POD
VODSTVOM GENERALA BURNSIDE-A"
2100 PRINT
2110 GOTO 2620
2120 IF A<>7 THEN GOTO 2180
2130 PRINT "ZA MURFREESBORO"
2140 IF A1=1 THEN GOTO 2620
2150 PRINT "31 PROSINAC 1862  "
2160 PRINT "JUG POD GENERALOM BRAGG OM
TIJESNO DOBIJA BITKU
2170 GOTO 2620
2180 IF A<>8 THEN GOTO 2250
2190 PRINT "KOD CHANCELLORSVILLE-A
2200 IF A1=1 THEN GOTO 2620
2210 PRINT "1-6 SVIBNJA 1863    JUG
POSTIZE SKUPU POBJEDU "
2220 PRINT "I GUBI JEDNOG OD SVOJIH
IZVANREDNIH GENERALA "
2230 PRINT "  STONEWALL  JACKSON-A "
2240 GOTO 2620
2250 IF A<>9 THEN GOTO 2320
2260 PRINT "ZA VICKSBURG
2270 IF A1=1 THEN GOTO 2620
2280 PRINT "4 SRPANJ 1863
2290 PRINT "JUG JE SKUPO PLATIO PAD
KOD VICKSBURGA JER SU UNIONISTI
2300 PRINT "OTVORILI IZLAZ NA MISSISSIPI
2310 GOTO 2620
2320 IF A<>10 THEN GOTO 2390
2330 PRINT "KOD GETTYSBURG-A
2340 IF A1=1 THEN GOTO 2620
2350 PRINT "30 LIPANJ 1863      OBNA
GRESKA GENERALA LEE-A KOD
2360 PRINT "GETTISBURG-A STAJALA JE
JUG JEDNE OD NAJVAZNIJIH BITAKA RATA
```

```
2380 GOTO 2620
2390 IF A<>11 THEN GOTO 2460
2400 PRINT "ZA CHICKAMAUGA-U "
2410 IF A1=1 THEN GOTO 2620
2420 PRINT "25 STUDENI 1863    NAKON
 STO JE JUG OPSJEDAO ARMIJU  "
2430 PRINT "GENERALA ROSENKRANTS-A TRI
 MJESECA GENERAL GRANT "
2440 PRINT "PROBIJA OPSADU "
2450 GOTO 2620
2460 IF A >12 THEN GOTO 2520
2470 PRINT "KOD CHATTANOOGA-E "
2480 IF A1=1 THEN GOTO 2620
2490 PRINT " 15 RUJNA 1863    ZBRKA U
  JMI KOD CHATTANOOGA-E "
2500 PRINT "DOVELA JE DO SKUPE POBJEDE
 JUGA
2510 GOTO 2620
2520 IF A >13 THEN GOTO 2590
2530 PRINT "ZA SPOTSYLVANIA-U "
2540 IF A1=1 THEN GOTO 2620
2550 PRINT "5 SVIBNJA 1864
2560 PRINT "PLAN GENERALA GRANTA DA
 IZOLIRA LEE-A POCINJE PROPADATI
2580 GOTO 2620
2590 PRINT "ZA ATLANTU "
2600 IF A1=1 THEN GOTO 2620
2610 PRINT "PROPAST KONFEDERACIJE
2620 PRINT
2630 RETURN
2640 PRINT "KONFEDERACIJA JE POBIJEDJEN
 A "
2650 GOTO 2670
2660 PRINT "UNIONISTI SU IZGUBILI RAT "
2670 PRINT
2680 PRINT "DOBILI STE  ;W;" BITAKA  A
 IZGUBILI ";L
2690 IF Y=5 THEN GOTO 2730
2700 IF W <= L THEN GOTO 2730
2710 PRINT "KONFEDERACIJA JE POBIJEDILA

2720 STOP
2730 PRINT "UNIONISTI SU DOBILI RAT   "
2740 STOP
-------------------------------- ------
```

Kraj

```
10 REM KRALJ  PREVEO ZUPAN STIPETIC
30 LET N5=0   LET N4=0
40 INPUT "UPUTE ";A$
50 IF A$="NE" THEN GOTO 420
70 PRINT
80 PRINT "CESTITAM
81 PRINT "UPRAVO STE POSTAVLJENI ZA
82 PRINT "PREMIJERA ZEMLJE NEODOVIJE
 MALE
90 PRINT "OTOCNE ZEMLJE OD 3000 MILJA
100 PRINT "VAS POSAO JE ODLUCIVATI O
 UDBINI ZEMLJE "
110 PRINT
120 PRINT "NOVCANA JEDINICA JE
 RALLOD SVAKI STANOVNIK MORA IMATI ";
130 PRINT "BAREM 100 RALLODA GODISNJE
 DA PREZIVI "
140 PRINT
150 PRINT "VAS BUDZET SE PUNI OD
 POLJOPRIVREDE TURIZMA
160 PRINT " I POREZA POLA VASE ZEMLJE
 JE OBRADIVO I MOZE SE PRODATI
170 PRINT " STRANOJ INDUSTRIJI KOJA
 DOVODI I PLACA VLASTITE RADNIKE "
180 PRINT
190 PRINT "SIJANJE I OBRADA ZEMLJE
 KOSTA IZMEDJU 10 I 15 RALLODA
210 PRINT "PO KVADRATNOJ MILJI
 GODISNJE,A DONOSE "
220 PRINT "IZMEDJU 45 I 65 RALLODA PO
 KVADRATNOJ MILJI GODISNJE
230 PRINT
240 PRINT "INDUSTRIJA KUPUJE ZEMLJU ZA
 95 DO 105 RALLODA PO "
251 PRINT "KVADRATNOJ MILJI "
260 PRINT " I PLACA POREZ OD 44 RALLODA
 PO KVADRATNOJ MILJI ZEMLJE "
270 PRINT "INDUSTRIJA POMAZE RAZVOJ
 TURIZMA CIM POSJEDUJE VISE OD
280 PRINT " 2 KVADRATNE MILJE PRIHODI
 OD TURIZMA PROPORCIONALNO
290 PRINT " RASTU DOK INDUSTRIJA IMA
 MANJE OD 20 KVADRATNIH "
300 PRINT "MILJA ZEMLJE
310 PRINT "INDUSTRIJA TAKODJER DONOSI
 I ZAGADJIVANJE "
320 PRINT "PROTIV ZAGADJIVANJA SE
 TREBA BORITI,A TO KOSTA
325 PRINT " 0 44 RALLODA PO JEDINICI
 ZAGADJENOSTI "
330 PRINT "ZAGADJENOST  E SMANJIVATI
 TURISTICKI PROMET I "
340 PRINT "POLJOPRIVREDNE PRINOSE
350 PRINT
360 PRINT "VASE POLJOPRIVREDNE PRINOSE
 MOZETE POVECATI
365 PRINT " OBRAZOVANJEM STANOVNISTVA
370 PRINT "VRIJEDNOST UPODA OVISI O
 SUMI UTROSENOJ "
380 PRINT "ZA OBRAZOVANJE U PROTEKLE
 GODINE
390 PRINT "10 RALLODA PO OSOBI  E
 NAJVECA KORISNA SVOTA
395 PRINT " ZA POTROSITI NA OBRAZOVANJE
400 PRINT
410 PRINT "VAS CILJ JE ZAVRSITI
 VAS" N4 " GODISNJI MANDAT
420 PRINT "SRETNO!"
430 LET M=INT(70000 +(RND*1000)-(RND*10
 00))
440 LET P=INT(500+(RND*10)-(RND*10)
450 LET L=2000
460 LET W=0:LET X2=0 LET E0=0 LET
 X1=0 LET E1=0
465 LET S9=0 LET S0=0 LET M5=0 LET S1=0
470 LET D1=INT(P/3)
480 LET L1=INT (RND*5)+10)
490 LET L0=INT(RND*10+95)
500 GOSUB 2270
510 PRINT "VI IMATE ";M;"  RALLODA
520 PRINT P;" STANOVNIKA  ";
530 IF W=0 THEN GOTO 550
540 PRINT W;" STRANIH RADNIKA
550 PRINT " I ";L;" KVADRATNIH MILJA
 ZEMLJE "
560 PRINT "OVE GODINE INDUSTRIJA CE
 KUPOVATI ZEMLJU PO
570 PRINT L0;"  RALLODA ZA KVADRATNU
 MILJU
580 PRINT "USJEVI I OBRADA ZEMLJE
 TRENUTACNO"
585 PRINT "KOSTAJU ";L1;" RALLODA PO
 KVDR MILJI
590 IF S1=0 THEN GOTO 610
600 PRINT "RAZINA ZAGADJENOSTI JE
 ";S1;" JEDINICA "
610 GOSUB 2270
620 INPUT "KOLIKO KVDR MILJA ZELITE
 PRODATI INDUSTRIJI";L2
640 IF L2 0 THEN GOTO 620
650 IF L2<(L-1000) THEN GOTO 730
660 PRINT "  RAZMISLITE MALO, VI IMATE
 SAMO"
670 PRINT L-1000   KVDR MILJA OBRADIVE
 ZEMLJE "
680 IF X1  0 THEN GOTO 620
690 PRINT "STRANA INDUSTRIJA NIJE
 ZAINTERESIRANA ZA POSJEDOVANJE
700 PRINT "ZEMLJU ZBOG VELIH TROSKOVA
 IZGRADNJE"
710 LET X1=1
720 GOTO 620
730 LET M=INT(M+L2*L0)
740 LET L=INT(L L2)
750 LET X9=INT(44*(2000-L  
760 LET M2=0
770 LET M3=0
780 LET M4=0
790 PRINT "KOLIKO RALLODA CETE
 RAZDJELITI "
800 INPUT " SVOJIM PODANI IMA  M1
820 IF M1 0 THEN GOTO  90
830 IF M1  = M THEN GOTO 860
840 GOSUB 2300
850 GOTO 790
860 LET M=INT(M-M1)
870 IF M=0 THEN GOTO 1220
880 INPUT "KOLIKO CETE KVDR MILJA
 ZASIJATI?  L3
900 IF L3 0 THEN GOTO 880
910 IF L3 <= P*2 THEN GOTO 940
920 PRINT "SVAKI STANOVNIK MOZE
 OBRADJIVATI SAMO 2 KVDR MILJE
930 GOTO 1000
940 IF L3 <= L 1000 THEN GOTO 970
950 PRINT "IMATE SAMO " -1000
 KVDR MILJA OBRADIVE ZEMLJE
960 GOTO 1000
970 LET M4=INT(L3*L1)
980 IF M4 <= M THEN GOTO 1040
990 GOSUB 2300
1000 PRINT "MOZETE FINANCIJSKI
 PODNIJETI ",INT(M/L1);
1010 PRINT " IMATE RADNE SNAGE ZA
 P*2,
1020 PRINT "  A IMATE ";L-1000
 KVDR MILJA
1030 GOTO 880
1040 LET M=INT(M-M4)
1050 IF M=0 THEN GOTO 1220
```

```
1060 INPUT "KO...ETE PL... 1 NA
OBRAZOVANJE "; M2
1080 IF M2<0 THEN GOTO 1060
1090 IF M2 <= M THEN GOTO 1120
1100 GOSUB 2300
1110 GOTO 1060
1120 LET M=INT(M-M2)
1130 IF M=0 THEN GOTO 1220
1140 IF L=2000 THEN GOTO 1220
1150 INPUT "KOLIKO DAJETE ZA BORBU
PROTIV ZAGADJIVANJA"; M3
1160 IF M3<0 THEN GOTO 1150
1180 IF M3 <= M THEN GOTO 1210
1190 GOSUB 2300
1200 GOTO 1150
1210 LET M=INT(M-M3)
1220 GOSUB 2270
1230 LET D0=0
1240 LET E2=M2/P
1250 LET D2=INT(P-M1/100)
1260 IF D2 <= 0 THEN GOTO 1290
1270 LET D0=D2
1280 PRINT D2;" STANOVNIKA JE POMRLO
OD GLADI"
1290 LET D3=INT((S1/100000 + 1+ 4*RND
*P)
1300 IF D3 <= 0 THEN GOTO 1330
1310 PRINT D3;" STANOVNIKA JE PREMINULO
PREVELIKOJ ZAGADJENOSTI"
1320 LET D0=D0+D3
1330 IF D0 <= 0 THEN GOTO 1450
1340 LET F=D0*9
1350 PRINT "BILI STE PRISILJENI
POTROSITI"
1355 PRINT F;" PALLODA ZA POGREBNE
TROSKOVE!"
1360 LET M=INT(M-F)
1370 IF M >= 0 THEN GOTO 1450
1380 PRINT "NEDOSTATAK NOVCA PRIMORAO
VAS JE NA PRODAJU ZEMLJE "
1390 LET T0=-INT((M-L0+1)/L0)
1400 LET L=L-T0
1410 IF L >= 1000 THEN GOTO 1440
1420 PRINT "NEMA DOVOLJNO ZEMLJE ZA
PRODAJU!"
1430 GOTO 2320
1440 LET M=M+T0*L0
1450 IF D0 200 THEN GOTO 2450
1460 LET P=P-D0
1470 LET D1=D1-D0
1480 IF D1<0 THEN GOTO 2460
1490 IF D2 = 2 THEN GOTO 1510
1500 IF M>500 THEN GOTO 2570
1510 IF L2=0 THEN GOTO 1570
1520 LET T0=INT(L2+L2+RND*2)
1530 IF W=0 THEN GOTO 1550
1540 LET T0=INT(T0+1*W)
1550 PRINT T0 " STRANIH RADNIKA JE
DOSLO NA OTOK I"
1560 LET W=W+T0
1570 LET T0=INT(((500-P)/10-D3 3 D2 5 *
75*(1+RND))
1580 PRINT ABS(T0) " VASIH DRZAVLJANA
JE ";
1590 IF T0<0 THEN GOTO 1620
1600 PRINT "DOSLO NA
1610 GOTO 1630
1620 PRINT "NAPUSTILO"
1630 PRINT " OTOK "
1640 LET P=P+T0
1650 IF P<W THEN GOTO 2530
1660 LET T0=0
1670 IF L=2000 THEN GOTO 1730
1680 LET T0=INT((S1/100000 )*L3
1690 IF T0 = L3 THEN GOTO 710
1700 LET T0=L3
1710 IF T0=0 THEN GOTO 1730
1720 PRINT "OD ZASIJANIH ";L3 "
KVDR MILJA,"
1730 PRINT " POZNJELI STE " L3-T0
KVDR MILJA "
1740 IF T0=0 THEN GOTO 1810
1750 IF X2 >= 2 THEN GOTO 1810
1760 PRINT " (ZBOG ",
1770 IF X2=0 THEN GOTO 1790
1780 PRINT "POVECANE ",
1790 PRINT "RAZINE ZAGADJENOSTI)"
1800 LET X2=X2+1
1810 LET T1=INT((39+RND*20)*(1+ 25*(E0+
E1)/20))
1820 PRINT " STO DONOSI ";INT(T1*(L3-T0)
);" PALLODA
1830 LET M=M+INT(T1*(L3-T0))
1840 LET E0=E1
1850 IF E2 <= 10 THEN GOTO 1870
1860 LET E2=10
1870 LET E1=E2
1890 IF T... THEN GOTO 2150
1900 IF ... - 26 THEN GOTO 1920
1910 LET ...-2A
1920 LET T0=INT(T0+500*(S2+ 5*RND
1930 LET T1= 1/100000
1940 IF T1 = 1 THEN GOTO 1960
1950 LET T1=1
1960 LET T1=T1*T0
1970 PRINT "ZARADILI STE ";INT(T0-T1
PALLODA OD TURIZMA
1980 LET M=M+INT(T0-T1)
1990 IF INT(T0-T1) = M5 THEN GOTO 2...
2000 IF S1 = S9 THEN GOTO 2130
2010 PRINT " TURISTICKI PROMET OPADA
JER ";
2020 LET T2=RND*10
2030 GOTO 2040+INT(T2/2)*20
2040 PRINT "RIBE UGIBAJU U PREZAGADJENO
M MORU "
2050 GOTO 2130
2060 PRINT "ZAGADJENOST ZRAKA SMANJUJE
PTICJA JATA
2070 GOTO 2130
2080 PRINT "MINERALNE KUPKE PROPADAJU
ZBOG ZAGADJENE VODE "
2090 GOTO 2130
2100 PRINT "NEUGODAN SMOG OBESHRABRUJE
UNLANIJE
2110 GOTO 2130
2120 PRINT "HOTELI DJELUJU PRLJAVO POD
TALOGOM CADJE
2130 LET M5=INT(T0-T1)
2140 LET S9=S1
2150 LET S2=(2000-L)*2-M3/ 44
2160 PRINT "ZARADILI STE ";S9 "
PALLODA OD POREZA NA INDUSTRIJU
2170 IF S2 >= 0 THEN GOTO 2220
2180 LET S1=INT(S1+S2/2
2190 IF S1 >= S0 THEN GOTO 2210
2200 LET S1=S0
2210 GOTO 2240
2220 LET S1=INT(S1+S2)
2230 LET S0=INT(S0+S2/10)
2240 LET N5=N5-1
2250 IF N5>0 THEN GOTO 480
2260 GOTO 2620
2270 PRINT
2280 PRINT
2290 RETURN
2300 PRINT "PREOSTALO VAM JE SAMO
";M;" PALLODA!"
2310 RETURN
2320 PRINT "IZBACENI STE S POLOZAJA I
SADA"
2330 PRINT "BORAVITE U ZATVORU "
2340 STOP
2350 PRINT D0 " STANOVNIKA JE UMRLO U
JEDNOJ GODINI!!!"
2360 PRINT "ZBOG OVOG OZBILJNOG
NEUSPJEHA STE NE SAMO
2370 PRINT "DEGRADIRANI I IZBACENI IZ
UREDA NEGO"
2380 LET T0=INT(RND*2 01)
2390 GOTO 2400+T0*20
2400 PRINT "VAM JE ISKOPANO I LIJEVO
OKO "
2410 STOP
2420 PRINT "STE TAKODJER DOSLI NA LOS
GLAS "
2430 STOP
2440 PRINT "STE TAKODJER JEDNOGLASNO
PROZVANI"
2445 PRINT "NACIONALNIM PREVARANTOM
BROJ 1 "
2450 STOP
2461 PRINT "PREKO TRECINE STANOVNISTVA"
2462 PRINT "JE UMRLO OTKAD STE VI
IZABRANI"
2470 PRINT "NA PREMIJERSKI POLOZAJ "
2472 PRINT "NAROD (ONO STO JE OSTALO)"
2480 PRINT "MRZI VAS IZ DNA DUSE!"
2490 GOTO 2500
2500 IF RND <= 5 THEN GOTO 2320
2510 PRINT "VI STE (S PRAVOM) UBIJENI "
2520 STOP
2530 PRINT "BROJ STRANIH RADNIKA JE
PREMASIO"
2540 PRINT "BROJ DOMACIH UROCANA "
2542 PRINT "VECINA SE POBUNILA I"
2550 PRINT "PREUZELA VLAST U ZEMLJI "
2560 GOTO 2500
2570 PRINT "PREOSTALO JE JOS NOVCA U
DRZAVNOJ"
2580 PRINT "KASI KOJI NISTE POTROSILI
2590 PRINT "DIO STANOVNISTVA JE
IZGLADNIO !"
2600 PRINT "PRISILJENI STE ILI
PODNIJETI"
2602 PRINT "OSTAVKU ILI IZVRSITI
SAMOUBOJSTVO
```

```
2610 STOP
2620 PRINT
2630 PRINT "BRAVO !!"
2640 PRINT "USPJESNO STE ZAVRSILI VAS "
2642 PRINT N4 "-GODISNJI MANDAT KAO PREMIJER."
2650 PRINT "OBAVILI STE POSAO TAKO DOBRO DA"
2660 PRINT "BI SE MOGLI OPET KANDIDIRATI"
2670 INPUT "DA LI TO I ZELITE? ";Q$
2690 LET N5=N4
2700 IF Q$="DA" THEN GOTO 470
2710 STOP
```

Dugme

```
110 REM DUGME
120 DEF FNM(X)=(X=8)*7+(X<>0 AND X<>8)*X+(X=0)*1
130 DEF FNN(X)=FNM(X+SGN(RND-.5))
140 INPUT "ZELITE LI PRAVILA ?";A$
150 IF A$="DA" THEN GOSUB 1000
220 LET B=INT(RND*7)+1
320 INPUT "STO MISLITE KOD KOGA JE DUGME ?";G
340 IF G=0 THEN GOTO 9999
350 IF G=B THEN GOTO 510
360 IF G=FNM(B+1) OR G=FNM(B-1) THEN GOTO 460
370 IF G>0 AND G<8 THEN GOTO 410
380 PRINT "ZANIMLJIVO - NEMA NITKO SA TIM BROJEM ";
390 PRINT "PONOVITE "
395 GOTO 320
410 PRINT G;"   TKO JA?"
420 PRINT: PRINT
440 PRINT "TKO GOD GA IMA ZADRZAO GA JE"
445 GOTO 320
460 PRINT G;"   IMA GA MOJ SUSJED"
470 PRINT: PRINT
490 PRINT "   ALI TKO GOD GA JE IMAO PROSLIJEDIO GA JE DALJE "
492 LET B=FNN(B)
495 GOTO 320
510 PRINT G;"   BRAVO   IMATE SRECE ! POGODILI STE !"
520 STOP
1000 PRINT
1030 PRINT "DUGME DUGME  TKO IMA DUGME ?"
1040 PRINT
1060 PRINT "SEDAM PRIJATELJA SJEDE U KRUGU"
1070 PRINT "A VI STE U CENTRU "
1080 PRINT
1100 PRINT "        1"
1110 PRINT
1120 PRINT "   7         2"
1130 PRINT
1140 PRINT "     TO STE"
1150 PRINT "6    VI    3"
1160 PRINT
1170 PRINT "   5     4"
1180 PRINT
1190 PRINT
1200 PRINT "NETKO IMA DUGME A VI TREBATE POGODITI TKO"
1220 PRINT "NAKON SVAKOG VASEG POKUSAJA,ONAJ TKO GA POSJEDUJE ";
1225 PRINT "MOZE (AKO HOCE) PROSLIJEDITI DUGME DALJE  ALI ";
1230 PRINT "SAMO SVOM SUSJEDU "
1240 PRINT
1250 PRINT "(AKO ZELITE PREKINUTI OTIPKAJTE 0)"
1260 PRINT
1270 PRINT "SRETNO !!"
1280 RETURN
9999 END
```

Morse

```
1000 REM MORSE ,NOVOSEL SINISA
1010 PRINT "OVAJ PROGRAM RADI SA MORSEOVOM ABECEDOM ";
1020 PRINT "PRETVARA TEKST U MORSEOVE ZNAKOVE I OBRATNO ";
1030 PRINT "REZIM RADA ODREDJUJE SE AUTOMATSKI ";
1040 PRINT "PREMA PRVOM ZNAKU UBACENOG TEKSTA ";
1050 PRINT "(MORSEOVE ZNAKOVE ODVAJAJTE SA / )"
1060 PRINT " IZVOLITE !"
1070 LET A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890 ,?-+/= "
1080 PRINT "A EVO I ZNAKOVA SA KOJIMA MOZETE RADITI": PRINT A$
1100 INPUT " ";B$
1110 IF LEN(B$)=0 THEN GOTO 9999
1120 IF B$(1 TO 1)="." THEN GOTO 1300
1130 IF B$(1 TO 1)="-" THEN GOTO 1300
1140 REM OVDJE SE PRETVARA TEKST U MORSEOVE ZNAKOVE
1150 FOR X=1 TO LEN(B$)
1160 FOR A=1 TO 44
1170 IF B$(X TO X)=A$(A TO A) THEN GOTO 1210
1180 NEXT A
1190 LET C$=B$(X TO X): LET C$(2 TO 5)="=UN?"
1200 GOTO 1250
1210 RESTORE
1220 FOR J=1 TO A
1230 READ C$
1240 NEXT J
1250 PRINT C$ "/ ";
1260 NEXT X
1280 GOTO 1100
1290 REM OVDJE SE PRETVARAJU MORSEOVI ZNAKOVI U TEKST
1300 LET N=1
1310 LET N2=0
1320 FOR J=N TO LEN(B$)
1330 IF B$(J TO J)="/" THEN GOTO 1370
1340 IF B$(J TO J)=" " THEN GOTO 1360
1350 LET N2=N2+1: LET L$(N2 TO N2)=B$(J TO J)
1360 NEXT J
1370 IF N2<>0 THEN GOTO 1410
1380 LET N=J+1
1390 IF N>LEN(B$) THEN GOTO 1100
1400 GOTO 1310
1410 RESTORE
1420 FOR I=1 TO 44
1430 READ D$
1440 IF L$(1 TO N2)=D$ THEN PRINT A$(I TO I);: GOTO 1380
1450 NEXT I
1460 PRINT "/";C$(1 TO N2) "=??";
1470 GOTO 1380
2000 DATA [illegible]
2010 DATA [illegible]
2020 DATA [illegible]
2030 DATA [illegible]
2040 DATA [illegible]
2050 DATA [illegible]
9999 END
```

Kodiranje

```
10 REM KODIRANJE
20 GOSUB 1000
30 K$="ZEBRA": REM GLAVNA SIFRA
40 LET L=0
50 INPUT "TEKST KOJI CE SE KODIRATI ";I$
60 FOR K=1 TO LEN I$)
70 LET L=L+1
80 IF L>LEN(K$) THEN LET L=1
90 POKE 32512,CODE K$(L TO L)
100 POKE 32513,CODE I$(K TO K)
110 O=USR 32500
120 PRINT O;" ";
130 NEXT K
140 STOP
1000 FOR I=0 TO 11
1010 READ A
1020 POKE 32500+I,A
1030 NEXT I
1040 RETURN
1050 DATA 58,0,127,71,58,1,127,168,79,6,0,201
```

Dekodiranje

```
10 REM DEKODIRANJE
20 GOSUB 1000
30 LET L=0
40 K$="ZEBRA": REM GLAVNA SIFRA
50 INPUT "KOLIKO ZNAKOVA JE BILO KODIRANO ?";S
60 FOR K=1 TO S
70 LET L=L+1
80 IF L>LEN(K$) THEN LET L=1
90 INPUT K;" SLIJEDECI =";J
```

```
100 POKE 32512 CODE K$(L TO L)
110 POKE 32513 J
120 O=USR 32500
130 PRINT "=  CHR$ O
140 NEXT K
150 STOP
1000 FOR I=0 TO 11
1010 READ A
1020 POKE 32500+I A
1030 NEXT I
1040 RETURN
1050 DATA 58,0 1,7 71 58,1 1,
 6,0,201
```

DVA MATEMATIČKA PROGRAMA

U ispisu oba programa sve su vrijednosti zaokružene na dvije decimale Ako su potrebni precizniji podaci, ispis se može tražiti direktno ili dodavanjem nove subrutine Oba programa vrše Gauss-Jordanove transformacije na redovima matrica, a uz male dopune mogu dati i općenitije rezultate Tako npr program »Sustav« rješava samo Cramerov sustav a može se proširiti za traženje općeg rješenja sustava linearnih jednadžbi dok program »Inverz« ujedno izračunava rang i determinentu matrice Zbog prezentacije brojeva u računalu može doći do pogrešnog rezultata ali samo u graničnim slučajevima kad je vrijednost determinente matrice A vrlo »blizu« nuli

Rješevenje eustava linearnih jednadžbi

```
1 REM SUSTAV LINEARNIH JEDNADBI
2 CLS
3 BORDER 4  PAPER 4
4 FOR i=0 TO 7
6 READ a
7 POKE USR  R +i,a
8 NEXT i
9 DATA BIN 00011000, BIN 00011200, BIN
000'1000, BIN 00000000, BIN 00000000,
BIN 00011000, BIN 00011000, BIN 00011000
10 CLS  INPUT  UNESITE RED SUSTAVA",n
15 IF n =0 THEN GOTO 10
20 DIM a(n,n)  DIM b(n)  DIM x(n)
25 IF n =6 THEN GOTO 110
30 FOR i=1 TO n
40 FOR j=1 TO n
50 INPUT "UPISITE KOEFICIJENTE PO
REDOVIMA", a(i,j)
60 PRINT AT 2*i, 2+5*(j-1); (INT
(100*a(i,j)))/100
70 NEXT j
80 INPUT  UPISITE SLOBODNI CLAN ,b(i)
90 PRINT AT 2*i,2+5*n  R
(INT(100*b(i)))/100
100 NEXT i
101 PAUSE 0
105 GOTO 300
110 FOR j=1 TO n
115 PRINT "UPISITE  ;j;  -TI STUPAC
116 PRINT
120 FOR i=1 TO n
130 INPUT a(i j)
140 PRINT (INT (100*a(i,j)))/100: PRINT
150 NEXT i
152 PAUSE 0
155 CLS
160 NEXT j
170 PRINT  UPISITE SLOBODNE CLANOVE :
PRINT
180 FOR i=1 TO n
190 INPUT b(i)
200 PRINT (INT (100*b(i)))/100: PRINT
210 NEXT i
215 PAUSE 0
217 CLS
300 INPUT  ZELITE  I  (D/N)  ,a$
310 IF a$="N" OR a$="n" THEN GOTO 500
  INPUT  KOEFICIJENT ILI SLOBODNI CLAN
(K/S)  b$
  IF b$="S" OR b$="s" THEN GOTO 350
  INPUT  MJESTO  ;i  ; ,j  ,"
  JEDNOST ";k
  LET a(i,j)=k
  CLS  GOTO 400
  INPUT  MJESTO  i  ;    VRIJEDNOST

  LET b(i)=l
400 FOR j=1 TO n
405 PRINT j  , -TI STUPAC
406 PRINT
410 FOR i=1 TO n
415 PRINT (INT (100*a(i,j)))/100: PRINT
417 NEXT i
420 PAUSE 0
425 CLS
430 NEXT j
435 PRINT "SLOBODNI CLANOVI"  PRINT
440 FOR i=1 TO n
445 PRINT (INT (100*b(i)))/100: PRINT
450 NEXT i
455 PAUSE 0
460 CLS  GOTO 300
500 GOSUB 6000
505 REM SVODJENJE NA TROKUTASTU MATRICU
510 FOR i=1 TO n-1
520 FOR k=i+1 TO n
521 IF a(i,i)=0 THEN GOTO 5001
522 LET rs=a(k,i)/a(i,i)
530 FOR j=1 TO n
540 LET a(k,j)=a(k,j)-a(i,j)*rs
550 NEXT j
560 LET b(k)=b(k)-b(i)*rs
570 NEXT k
575 GOSUB 6000
580 NEXT i
5000 REM RJESAVANJE
5001 IF a(n,n)  0 THEN GOTO 5010
5002 IF b(n) 0 THEN GOTO 5004
5003 PRINT AT 1 6; FLASH 1  SUSTAV NEMA
RJESENJA  GOTO 7000
5004 PRINT AT 1 1  FLASH 1  SUSTAV IMA
BESKONACNO RJESENJA": GOTO 7000
5010 LET x(n) b(n)/a(n,n)
5020 FOR i=n 1 TO 1 STEP -1
5030 LET k(i)=b(i)
5040 FOR j i+1 TO n
5050 LET k(i) k(i) a(i j)*x(j)
5060 NEXT j
5070 LET x(i)=k(i)/a(i,i)
5080 NEXT i
5090 FOR i 1 TO n
5100 PRINT  X( ,i; ) ";x(i)
5110 NEXT i
5120 GOTO 7000
6000 FOR q=1 TO n
6010 FOR w=q TO n
6020 IF a(w,q) 0 THEN GOTO 6080
6030 FOR e=1 TO n
6040 LET c a(q,e)  LET a(q,e) a(w,e)
LET a(w,e)=c
6050 NEXT e
6060 LET z=b(q)  LET b(q) b(w): LET
b(w)=z
6070 GOTO 6090
6080 NEXT w
6090 NEXT q
6100 RETURN
7000 INPUT "ZELITE ZADATI NOVI SUSTAV
(D/N)   ,a$
7010 IF a$="D" OR a$="d" THEN GOTO 10
```

```
7020 PRINT AT 21,11; FLASH
1;"DOVIDJENJA": PAUSE 100: PRINT USR 0
```

Izračunavanje inverza matrice

```
5 BORDER 6: PAPER 6
10 CLS: INPUT "UNESITE RED MATRICE" n
15 IF n =0 THEN GOTO 10
20 DIM a(n,2*n)
25 IF n >7 THEN GOTO 110
30 FOR i=1 TO n
40 FOR j=1 TO n
50 INPUT "UPISITE ELEMENTE PO REDOVIMA" a(i,j)
60 PRINT AT 2*i,2+5*(j-1);(INT
(100*a(i,j)))/100
70 NEXT j
80 NEXT i
90 PAUSE 0
100 GOTO 200
110 FOR j=1 TO n
115 PRINT "UPISITE ";j;". TI STUPAC"
120 PRINT
125 FOR i=1 TO n
130 INPUT a(i,j)
140 PRINT (INT (100*a(i,j)))/100: PRINT
150 NEXT i
152 PAUSE 0
155 CLS
160 NEXT j
200 INPUT "ZELITE LI IZMJENE (D/N)",b$
210 IF b$="N" OR b$="n" THEN GOTO 400
220 INPUT "MJESTO ";i;",";j;",
    VRIJEDNOST ";t
230 LET a(i,j)=t
240 CLS
300 FOR j=1 TO n
310 PRINT j;". TI STUPAC"
320 PRINT
330 FOR i=1 TO n
340 PRINT (INT (100*a(i,j)))/100:PRINT
350 NEXT i
360 PAUSE 0
365 CLS
370 NEXT j
380 GOTO 200
400 FOR i=1 TO n
410 FOR j=n+1 TO 2*n
420 LET a(i,j)=0
430 NEXT j
440 LET a(i,n+i)=1
450 NEXT i
460 GOSUB 8000
500 REM SVODJENJE NA DIJAGONALNU MATRICU
510 FOR i=1 TO n-1
520 FOR l=i+1 TO n
521 IF a(i,i)=0 THEN GOTO 7000
522 LET rs=a(l,i)/a(i,i)
530 FOR j=1 TO 2*n
540 LET a(l,j)=a(l,j)-a(i,j)*rs
550 NEXT j
570 NEXT l
575 GOSUB 8000
580 NEXT i
585 GOSUB 8000
590 FOR i=n TO 2 STEP -1
600 FOR l=i-1 TO 1 STEP -1
605 IF a(i,i)=0 THEN GOTO 7000
610 LET rs=a(l,i)/a(i,i)
620 FOR j=1 TO 2*n
630 LET a(l,j)=a(l,j)-a(i,j)*rs
640 NEXT j
650 NEXT l
660 NEXT i
700 FOR i=1 TO n
710 LET t=1/a(i,i)
720 FOR j=1 TO 2*n
730 LET a(i,j)=a(i,j)*t
740 NEXT j
750 NEXT i
6200 IF n>7 THEN GOTO 6300
6210 CLS
6220 FOR i=1 TO n
6230 FOR j=1 TO n
6240 PRINT AT 2*i,2+5*(j-1);(INT
(100*a(i,n+j)))/100
6250 NEXT j
6260 NEXT i
6270 PAUSE 0
6280 GOTO 7100
6300 FOR j=1 TO n
6310 PRINT ;j;". TI STUPAC"
6320 PRINT
6330 FOR i=1 TO n
6340 PRINT (INT (100*a(i,n+j)))/100:
PRINT
6350 NEXT i
6360 PAUSE 0
6361 CLS
6370 NEXT j
6380 GOTO 7100
7000 PRINT AT 5,7; FLASH 1; "MATRICA JE
SINGULARNA"
7010 PAUSE 0
7020 CLS
7100 INPUT "ZELITE LI ZADATI NOVU MATRICU
(D/N)",a$
7110 IF a$="D" OR a$="d" THEN GOTO 10
7120 PRINT USR 0
8000 FOR q=1 TO n
8010 FOR w=q TO n
8020 IF a(w,q)<>0 THEN GOTO 8080
8030 FOR e=1 TO 2*n
8040 LET r=a(q,e): LET a(q,e)=a(w,e):
LET a(w,e)=r
8050 NEXT e
8070 GOTO 8090
8080 NEXT w
8090 NEXT q
8100 RETURN
```



Djeca uče elemente programske tehnike uz pomoć 'kornjače' koja na velikom komadu papira crte slike pod kontrolom LOGO jezika

Čovjek ne može dugo živjeti u svijetu kompjutora a da se kad-tad ne sudari sa pojmom strukturiranog programiranja jer je previše strasti uloženo u raspreve treba li djecu — i ne samo djecu — učiti tom načinu programiranja ili ne treba

Najveći je problem u BASICu Lako se uči kažu jedni a drugi tvrde kako je BASIC iluzija Istina je da ga se lako uči ali je on kao mlaka u nju se lako ulazi može se bezbrižno hodati po njoj ali se ne može plivati — preplitko je

Jezici poznati kao strukturirani — teže se uče jer se ne vladaju kao obični nekompjutorski jezici Najznačajnije im je svojstvo što se mogu dograđivati to jest sastavljati od malih logičkih blokova nazvanih procedurama, koje su na najnižoj razini nalik na one objedovne potprograme što smo ih naveli na str 56—57 BASIC vas sili da se ponašate poput samouka majstora koji sam gradi vikendicu doprema na gradilište sav materijal s kojim raspolaže i sastavlja kućicu od onoga što mu najprije dopadne ruku kako mu se prohtije bez plana i nacrta

Ali što je dobro za vikendicu nije dobro za katedralu Onaj koji gradi katedralu treba postupati kao graditelj a ne kao sam svoj majstor' Treba najprije od kamenih blokova sastaviti stupove i lukove od stupova i lukova svodove od svodove crkvene lada a od njih — katedralu Graditelj vjerojatno nikada ne bi sagradio katedralu kad bi u svakoj fazi gradnje bio prisiljen razmišljati o svakom pojedinom kamenu a to je baš ono što BASIC traži

Ima vise strukturiranih jezika i mogu se podijeliti u dvije skupine u prvu spadaju FORTRAN Pascal C' i ALGOL, poznati kao Von Neumannovi jezici nazvani po istaknutom matematičaru Von Neumannu cije je ime vezeno uz rani razvoj kompjutora U drugu skupinu spada cijeli niz procesnih jezika kao što su LISP LOGO i PROLOG

LOGO je najviše poznat po strastvenom zagovaranju njegovih propagatora Osnovni smisao tog jezika je namjera da se kompjutor prilagodi nastavi u školskoj učionici pa tako jedna jednostavna ali precizne naprava (kornjača) puže po velikom komadu papira rasprostrtom po podu i crta olovkom Kablom je povezana s kompjutorom i reagira na naredbe poput ovih

FORWARD X (idi naprijed za duljinu X)
RIGHT A (skreni desno za A stupnjeve)
DRAW (spusti olovku i crtaj)

Program se definira jednostavnim riječima da bi se nacrtao kvadrat treba izdati naredbu HOOK (kuka) što znači da kornjača treba otpuzati X jedinica FORWARD (naprijed) i tada se okrenuti RIGHT (desno) Ako se ovo ponovi četiri puta nacrtan je SQUARE (kvadrat)

Nakon nekog vremena uporan LOGO programer (isto je s LISP i FORTH jezicima) raspolagat će s velikim brojem verbalnih naredbi koje se mogu bez teškoca međusobno kombinirati

Jednostavno rečeno, kompjutorski jezik je sredstvo kojim korisnik komunicira s kompjutorom služeći se preciznim naredbama. Takav jezik mora biti kompromis između onoga što je s jedne strane kompjutoru prepoznatljivo, a s druge čovjeku razumljivo.

Prvi kompjutorski jezik bio je strojni (mesinski) kôd (vidi str. 80—85.). Bio je to vrlo neugodan i težak jezik, jer je decimelna aritmetika u binarnom kodu tako veliki psihički napor da je ostavio duboke posljedice na ljudima koji su se njime bavili. Ali na sreću, danas je u radu s kompjutorima poznavanje binarne aritmetike isto tako nepotrebno kao što je astronavigacija nepotrebna avionskom putniku.

Radeci u strojnom kôdu, prvi su programeri vrlo brzo primijetili de stalno ponavljaju neke krace dijelove kôda, pa im je pela na pamet sjajna zamisao da naprave ono sto danas zovemo potprogramime (str. 56—57) i makro naredbama. Makro naredba je zapravo izvjesni dio koda koji služi kao jedinstvena naredba. Na primjer, da bismo zbrojili dva decimalna broja uključujemo makro naredbu. O tome nakon nekog vremena više ne mislimo kao o kôdu, postaje to jednostavno makro naredba 'ADD'. S vremenom će se sve funkcije pretvoriti u kratice i kompjutor će raspolagati bibliotekom makro naredbi od kojih se može sastaviti program poput ovoga:

ENTER NUMBER (unesi broj)
ADD NUMBER 6 ANSWER (dodaj broj 6 odgovori)
PRINT ANSWER (napiši odgovor)

ENTER, ADD i PRINT su makro naredbe, a varijable koje ih upotpunjavaju — NUMBER za ENTER, NUMBER 6 i ANSWER za ADD i ANSWER za PRINT — stoje uvijek na određenom mjestu i zovu se 'argumenti'.

Još uvijek to nije viši programski jezik, ali je na najboljem putu da postane. Programer koji mnogo radi u strojnom kôdu brzo će shvatiti da postoje dvije razine programiranja: programske linije pisane u strojnom kôdu na uobicajeni način i veci dijelovi koda koje upotrebljava kao gotove samostalne jedinice.

To zahtijeva dve stupnja rada: prvo, kodiranje ili asembliranje neredbi, a onda povezivanje pojedinih blokova u cjelovit program. To je shema koja se još uvijek primijenjuje u tradicionalnim jezicima kao što su Assembler, FORTRAN, ALGOL i COBOL.

U početku je ALGOL bio standardno sredstvo zapisivanja matematičkih formula, pa uopce nije bio smatran kompjutorskim jezikom. Kasnije se netko dosjetio da je ALGOL zapravo kompjutorski jezik, ako se njime mogu pisati makro naredbe koje odgovaraju matematičkim instrukcijama i rade na sukladan način. Shvativši to, trebalo je još samo načiniti posebni program koji će pročitavši nazive makro naredbi vaditi jednu po jednu iz kompjutorove biblioteke i unositi ih u kôd. Taj je program nazvan 'kompajler' (compiler — kompilator, sastavljac).

Ali iza kompajlera se krije zamka koju je korisnicima podmetnulo tržiste. Naime, razlike koje postoje između sve većeg broja računala što se dan za danom proizvode u svijetu, mogu se vrlo lako prikriti kompajlerom. Mnogi mali evropski proizvođaci koji su proizvodili inkompatibilni hardver, htjeli su (a hoce još i danas) kompajlerom zabašuriti razlike između svojih računala. U Americi su međutim stvari stajale drugačije: tamosnjom scenom vladao je svemoćni IBM i sva su računala bila kompatibilna.

Nakon 1950. pojavio se izvjestan broj programa za pisanje strojnih kodova pod zajednickim nazivom 'autokôdovi' (autocodes) i iz njih su nastala dve jezika: FORTRAN i PL/1. FORTRAN (Formula Translation — prevođenje formule) je služilo sasvim dobro i još danas je u upotrebi. PL/1 (Programming Language 1 — programski jezik 1) je zamisljen kao temeljno rješenje svih jezičnih problema, ali jer se stvarao malo po malo, umjesto da nastaje kao jedinstvena arhitektonska cijelina, nikada nije postao zaista populeran.

Iz autokodova se razvio i COBOL (Common Business Oriented Language — opći komercijalni jezik). Izumio ga je mornarički kapetan Grace Hopper i čini se da ga od prvoga dana nitko nije volio. Namjera je bila da se stvori jezik poput govornog engleskog, ali nakon nekog vremena ta je zamisao napuštena i zamijenjena nadom da će programe razumijevati oni koji govore engleski. Nastalo je nesto poput ovoga:

```
000480 IF CRT—STOCK—CODE=SPACE
GO TO END—IT
000490 IF CRT—UNIT—SIZE NOT NUMERIC
GO TO CORRECT ERROR
000500MOVE CRT—PROD—DESC1 TO
PRODUCT—DESC
```

Prеđimo sada na ALGOL (Algorithmic Language — algoritamski jezik) koji je pomogao da se usavrši FORTRAN, u smislu mnogo elastičnijih potprograma s vlastitim, izvana nevidljivim internim varijablama. Međutim, čistunci što su se nalazili u komisiji koja je sastavljala ALGOL odbijali su bilo kakav način kojim bi programi pisani u tom jeziku primali podatke od vanjskog svijeta i davali rezultate nestrpljivim korisnicima. Smatrali su da je program nešto što se zbiva potpuno unutar kompjutora, bez ikakva odnosa s vanjskim svijetom, pa je tako ovaj nesretni propust u priličnoj mjeri omeo razvoj ALGOLa.

Iako ALGOL nikada nije imao nekog većeg uspjeha u odnosu na broj programa, u teorijskom smislu predstavljao je ipak veliki korak naprijed. Omogucio je ostvarenje 'rekurzije', matematičkog pojma kojim se u FORTRANu služe programeri kada jednim potprogramom pozivaju drugog. Ali u FORTRANu nije moguce izvesti da potprogram poziva i sama sebe

ALGOL je uklono i ovo ograničenje tako da se moža postaviti neki potprogram, sakriven negde u gomili kôda koji ćo inicirati samoga sebe koliko god puta bude trebalo Rekurzija za potprograme znači isto ono što potprogram znači pojedinim naredbama (str 56—57) omogućava da ih upotrijebimo koliko god puta nam zatreba

Slijedeći korak naprijed bio je BASIC Poznala je mana kompilatorskih jezika da troše mnogo vremena za kompiliranje Ako program ne radi a to se najčešce događa treba se vratiti na početak izvornome kôdu promijeniti ga i iznova kompilirati Na početnika to može djelovati obeshrabrujuce pa je zato bila osnovna zamisao da se u BASICu svaka pojedina programska linija odmah prevede i izveda svaki put kada se preko nja pređe To je vrlo korisno jer se može napisati samo dio programe poljerati ga pa ako nešto nije u redu odmah i popraviti To se svojstvo pokazalo tako značajnim da je BASIC — izraženo brojem računala u koja je ugrađen — ubrzo zavladao svijetom kompjutora All je ujedno to i razlog da BASIC troši veci dio svog radnog vremena na prevođenje ili interpratiranje (zato se softver za tu operaciju naziva 'interpreter') onoga što je korisnik napisao kao programske naredbe a samo manji dio na izvršavanje tih naredbi

Interpreter BASICa vrlo je polagan jer stalno mora obavljati ogroman posao ali zato postoji kompajler u kojem BASIC radi oko desat puta brže nego pomoću interpretera I asembler koji ća istu stvar obaviti još pet puta brže

Pascal jezik koji je osnovao Švicarac Nikolaus Wirth pokušaj je ujedinjenja najboljih osobina ALGOLa i BASICa U Pascalu treba najprije deklarirati sve varijable — ne postupno tokom programa kao u BASICu već na početku — a dok se program vrti obavlja se mnoštvo provjera verijabli Potprogrami su mnogo prikladniji nego u BASICu jer se mogu pozivati rekurzivno, što nije moguce u većini verzija BASICa

Pošto su ti visi programski jezici već neko vrijeme bili u širokoj upotrebi došlo se do zaključka da se još uvijek trosi previše vremena na strojni kôd asembler i na sve ostale stvari koje zadaju glavobolju programerima Osim toga viši programski jezici ne mogu se dovoljno komprimirati da bi stali u raspoloživi memorijski prostor a kad su vec ovdje, ne vrte se dovoljno brzo

Da bi se ubrzalo izvođenje programa, stvoreni su novi jezici koji su zapravo poboljšane verzije strojnog kôda Iako teži od uobičajenih viših programskih jezika oni se vrte brže i komprimiraju na manji memorijski prostor Jedan od njih je i FORTH neobičan jezik koji na primjer traži da se piše '3 4 +', umjesto '3 + 4' slično kao na nekim kalkulatorima

Još jedan jezik kompaktan i brz ali na račun jednostavnosti je C Nalik je Pascalu osim što ima manje elemenata osiguranja i direktan pristup procesorovim registrima Program je poput ovoga.

```
compare (p1 p2 n)
      char*p1,*p2
      Int n [
         register Int m
         for (m=0 m<n&&*p1==*p2&&
            *p1!='/0'++p1 ++p2++m),
         return (m==n//n*p2==*p1),]
```

Bilo bi teško tvrditi da je na prvi pogled razumljiv ali da je efikasan u to sam uvjeren

Postoji još jedna grana obrade podataka sasvim drugačija od klasične kompjutorske discipline To je tzv umjetna inteligencija (AI — Artificial Intelligence) Ljudi koji se njome bave rade nešto sasvim drugo nego matematičari i inženjeri što se bave klasičnim programskim jezicima Kod umjetne inteligencije počinje se na primjer rečenicom Uzmi nož Iz te je rečenice jasno što se hoće ali nije jasno kako to treba izvesti Na koji način ljudsko oko i um vide nož na koji ga način treba uzeti i kako razlikovali od nekog drugog predmeta? Da bi tome doskočili izmislili su jezik LISP (List Processing — obrada popisa) koji uvijek počinje s nekom velikom općenitom idejom na primjer

SMISAO SVEMIRA (BOGOVI LJUDI)

Nakon toga treba pokušati ideju raščlaniti dopunskim idejama kao

PRIRODA (ČOVJEK SMRTAN)
PRIRODA (BOG, BESMRTAN)

Treba biti uporan pa na isti način dalje nastaviti s novim idejama (smrtan besmrtan) koje bi mogle biti relevantne za temu i nadati se u dogledno vrijeme programu koji će objasniti smisao svemira

Čini se na prvi pogled da je to veliki napredak u odnosu prema krutosti Von Neumannovih jezika ali upravo elastičnost LISPa (i njemu sličnih jezika) prijeti da vrlo brzo zaguši programera s mnoštvom riječi-naredbi koje se lako izmišljaju, ali još lakše zaboravljaju Konvencionalni jezici imaju skučen, ali jasno definiran rječnik što je svakako manje uzbudljivo ali vrlo praktično Vjerojatno je upravo ta činjenica razlogom što jezici tipa LISP nikada nisu uhvatili dubljega korijena u općoj obradi podataka

Postoji mnoštvo potencijalnih kupaca kompjutora koji smatraju da proces programiranja treba učiniti pristupačnijim da se, barem prividno otkloni učenje nečega sasvim novog Potaknut takvim zahtjevima upravo se pojavljuje novi trend što ga užurbano forsiraju velike kompanije Pionir je tvrtka Xerox s računalom Star a nastavak Appleova Lisa (vidi str 46—47)

STROJNI KÔD I STRUKTURA PODATAKA

Sade je već sasvim jasno da je svaki viši programski jezik tek sredstvo za pripremanje zalogaja strojnoga koda kojim se hrani procesor. Da bismo potpuno shvatili suštinu strojnoga koda, moramo procesoru posvetiti više pažnje nego što je učinjeno na str. 16—17 jer on u stvarnosti ima više od tri registra za pohranu i manipulaciju podataka i mnogo više od tri jednostavne naredbe.

Pogledajmo stoga procesor Z80 ugrađen u veći dio suvremenih kompjutora. To je zapravo dvostruki procesor s dvije garniture registara i programer se može prebacivati s jednog na drugi, upravo kao što vješt tkalac može istodobno raditi na dva razboja. U svakoj garnituri ima posebni 1-bajtni registar, nazvan akumulator (A), tri 2-bajtna registra, četvrti namijenjen čuvanju adresa i još jedan posebni registar (F) za zastavice.

Zastavica ili flag je posebna kompjutorska dosjetka i ništa tome slično ne postoji izvan svijeta kompjutora. To je signal da se nešto *jest ili nije dogodilo*. Na primjer, Z80 ima posebnu naredbu za usporedbu bajtova, CP B. Tom se naredbom komparira bajt u B registru sa bajtom u A registru. Ako su jednaki, zastavica 0 (ili flag 0) — to je drugi bit u bajtu zastavice — postavlja se na 0. To vrijedi i za druge naredbe, na primjer za naredbu jump (skoči). Programer može narediti skok na neki drugi dio programa ako su bajtovi u registrima A i B jednaki, a ako nisu, može se vratiti, uvrstiti novi bajt u B registar i ponovo provjeriti jednakost.

Garnitura komandi Z80 procesora, pored već spomenutih funkcija zbrajanja, oduzimanja i kompariranja, koje su najvažnije, sadrži još i ove mogućnosti:

puni bajtove iz memorije u registre i obratno,

međusobno izmjenjuje sadržaje registara

kopira podatke iz jednog dijela memorije u drugu

pretražuje određeni blok memorije u potrazi za nekim bajtom

izvodi logičke funkcije I, ILI, isključivo ILI (XOR, vidi str. 20—21) na dva bajta u registrima;

obavlja neke prilično egzotične funkcije koje omogućavaju decimalnu aritmetiku s binarnim sredstvima

preskače s jedne programske linije na drugu što se može kontrolirati sadržajem registra. To na primjer omogućuje da se pronađe slovo a u memorijskom prostoru pomoću određene instrukcije, pa ako ga nađe izvršava jedan dio programa, a ako ne — drugi.

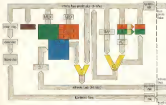
može rotirati bitove lijevo ili desno po registru, provjeriti svaki i postaviti ga na '0' ili '1'

može primati bitove s ulazne porte i slati ih na izlaznu

može pozivati potprograme i vraćati se glavnom programu

može prekinuti izvođenje tekućeg programa i skočiti na neku drugu programsku liniju

Na prvi pogled mnoge stvari u ovom popisu čine se suvišne ili barem nerazumljive. Ali činjenica je da je programiranje na razini asemblerskog jezika vrlo polagano i naporno, pa kad bismo htjeli problem potanko objasniti, zauzelo bi to prostor jedne čitave knjige. Jedino što možemo učiniti je da se makar i površno dotaknemo te najfundamentalnije razine svakog računala.



Pojednostavnjena shema procesora Z80 koji u stvari sadrži dva kompletna procesora, te su zato na slici registri A do L prikazani dva puta. Prebacivanje s jedne na drugu grupu registara kontrolira se programski. 8-bitni akumulatorski registar (A) povezan je s aritmetičkom logičkom jedinicom (ALU) koja svoj drugi ulaz dobija preko data busa, a izlaz šalje busu i flag registru (F). Ostali registri organizirani su u parovima, u nekim slučajevima mogu djelovati poput 16-bitnih. IX i IY su 16-bitni registri koji služe za adresiranje data, a SP je 16-bitni pokazivač položaja (stack pointer). PC upućuje na sljedeću naredbu koju treba izvršiti. Za privremenu pohranu podataka služe bafer (bufferi).

Asembler

Kao što smo vidjeli na str. 24—25, program je smješten u jedan dio memorije, a podaci (data) koje program obrađuje u drugi. Pokušajmo sada sastaviti mali program koji će po memoriji tražiti ime Petar. Može nam to zatrebati kao dio programa za obradu teksta (word processor) jer bismo možda htjeli Petra prekrstiti u Ivana.

Ima nekoliko načina da se izvrši ovaj zadatak, ali najlakše je početi s lijevog kraja dijela memorije koji treba pretražiti i testirati svako slovo da li odgovara slovu P. Ako ga nađemo, treba tražiti sljedeće slovo 'e', kad se i ono nađe, sljedeće koje se testira je 't'; ako slova 't' nema, procesor se vraća natrag na P i počinje ispočetka.

Većina naredbi što ih Z80 prepoznaje sastoji se od jednoga bajta koji sadrži naredbu i još dva bajta u kojima su sadržane adrese. Naredba skoči na memorijsku lokaciju m n' glasi 11000011 (m) (n). Čak i najbolji profesionalni programer ne može upamtiti na stotine ovakvih kôdova, pa se stoga strojni kôd obično piše u asembleru, jeziku kojim se bajtovi zamjenjuju

mnemoničkim kôdovima Mnemonički kod za skoči na m n, glasi JP m n (JP je mnemonik za jump — skoči)

Većina asemblerskih programa ne specificira memorijske adrese One su u tzv relociranom formatu, što znači da ih se može bez posljedica staviti u bilo koji dio memorije što je vrlo korisno jer je kompletni program uvijek sastavljen od mnogo malih segmenata strojnoga koda od kojih se svaki piše i testira zasebno Mjeste u memoriji gdje se segmenti testiraju i mjesta na kojima konačno rade mogu biti sasvim različita

Segmente povezuje "linker" program što povezuje dijelove kôda u kompaktnu cjelinu slobodno prilagođavajući memorijske lokacije

Ako smo shvatili ova uvodna objašnjenja, možemo pokušati da potražimo Petra! Pretpostevimo da se Petar nalazi u području memorije čiji prvi bajt ima naziv IME a da se tekst koji tražimo nalazi u području pod imenom TEKST

Primjer programa u strojnom kôdu

```
DSEG
NAME      DB     Peter
TEXT      DB     Treba nekoga poslati u Pariz —
                 vjerojatno Petra./
CSEG
START     LD     HL TEXT
LO        LD     DE NAME
          LD     A (DE)
          LD     B A
L1        LD     A (HL)
          CP                   potraži svršetek
                               teksta
          JR     Z NOTFND      skoči ako je
                               gotovo
          CP     B             traži sljedeće
                               slovo
          INC    HL
          JR     NZ L1
          PUSH   HL
L2        INC    DE
          LD     A (DE)
          CP
          JR     Z FOUND
          CP     (HL)
          INC    HL
          JR     Z L2
          POP    HL
          JR     L3
NOTFND                         ovdje skreće
                               program koji se
                               izvodi ako Petar
                               nije nađen
FOUND                          ovdje skreće
                               program koji se
                               izvodi ako je Petar
                               nađen
```

Kao što se vidi to je dosta komplicirano i teško je vjerovati da bi netko poželio da cijeli život radi u strojnom kodu Pametan čovjek će raditi u strojnom kodu samo kada zaista nema drugog izlaza kad treba program koji zauzima malo memorijskog prostora i radi vrlo brzo

Strojni kod se obično koristi za sistemski softver ili one dijelove programa koji se često upotrebljavaju i koji bi napisani u nekom od viših programskih jezika previše usporavali rad računala

Profesionalni proizvođači softvera ne vole strojni kôd, preskup je teško ga je održavati (što znači da ga je teško popraviti kad se negdje pojavi greška) a redovito ga treba potpuno preraditi za neko drugo računalo ili kada pak izvorni programer promijeni poslodavca



Pointeri

Najjednostavnije rečeno pointer je adresa lokacije na kojoj je nešto pohranjeno Varijable u BASICu smještene su na listi imena od kojih svaka ima uza se po jedan pointer koji upućuje na neki drugi dio memorije gdje je pohranjena njena vrijednost Na primjer na crtežu u dnu stranice pointer varijable FRUITS pokazuje na voće apple Pointeri se ekstenzivno upotrebljavaju u suvremenom programiranju pa na primjer jezik C koji se upotrebljava za najozbiljnije programiranje mnogo manipulira pointerima (Znak * u primjeru C jezika na str 79 označava pointer sljedeće varijable) Može nastati zbrka ako se pointerom pokazuje na drugi pointer ali ljepota je u tome da se pointerom može usmjeriti program na razna mjesta memorije bez premještanja podataka na mjesto operacije Programiranje s pointerima nalik je noćnom lovu na zečeve uz pomoć snažnih reflektora

Slaba je strana upotrebe pointera za označavanje podataka koje želimo obrađivati što se memorija — a i disk — može napuniti podacima na koje ne pokazuje niti jedan pointer (kao što je slučaj na slici na sljedećoj stranici gdje na smjer u kojem se nalazi lemon ne pokazuje niti

Neki BASICi memoriraju tekst pomoću popisa pointera i duljine teksta Pointer (pokazivači) 111 označava položaj stringa apple i njegovu duljinu 5 slova Taj način olakšava pohranu i pronalaženje stringova različitih duljina



jedan pointer). Da bi se spriječilo punjenje memorije takvim otpacima, programi s pointerima moraju imati neku 'komunalnu službu', tj. potprograme koji patroliraju po memoriji poput smetlara, kontroliraju svaki podatak, pa ako nađu takav na koji nije upućen niti jedan pointer, brišu ga, a prostor koji je zauzimao vraćaju na popis slobodnih memorijskih lokacija kako bi ga slijedeći potprogram mogao koristiti.

Stak

U svijetu kompjutora pojam 'stak' (engl. stack — stog, slog, ili složaj, kako predlažu neki domaći autori. Op. prev.) ima sasvim osobito značenje. To je memorijski prostor gdje se pohranjuju podaci fiksne dužine — 2 bajta u 8-bitnim računalima — koji sadrže memorijske adrese, a svaki ima pointer što pokazuje na slijedeću praznu adresu u posebnom registru procesora. Naravno da i program može imati svoj vlastiti stak. Podatak se može uložiti (push) u stak i izvaditi (pop) iz njega, pa zato većina asemblerskih programa ima komande PUSH i POP'. Običaj je da se za stak na vrhu programskog prostora rezervira nekoliko stotina bajta memorije. Glavna primjena staka je u potprogramima. Na kraju potprograma stak postavlja adresu na koju program treba da se vrati.

Aritmetika i plivajuća točka (čitatelji se slabim srcem mogu ovo poglavlje preskočiti)

Ako napišemo programsku naredbu koja sadrži poruku C = 2 + 3, tražimo od računala da to izračuna i da rezultat pohrani pod imenom varijabla C. Za kompjutor izraz 2 + 3 predstavlja string triju ASCII simbola 00110010 00101011 i 00110011. Postoji dio programa nazvan 'parser' koji analizira string i prepoznaje ga u obliku 'broj', 'operator', 'broj'. Operator + kaže da treba učiniti određenu operaciju pomoću odgovarajućeg dijela strojnog koda. Svaki od operatora — * i / (oduzimanje, množenje i dijeljanje) naređuje različitim dijelovima strojnoga kóda da obava svoj specifični posao.

Neće nam biti previše teško zbrojiti 2 i 3, jer se oba broja mogu pretvoriti iz svojih ASCII kódova direktno u bajtove 00000010 i 00000011, a ugrađena procesorova naredba za zbrajanje lako će obaviti zadaću. Svaki 8-bitni procesor ima dvostruke registre pa može obrađivati najviše 65 536 brojeva. Ali postoji, dakako, i operacija oduzimanja, pa se mogu pojaviti i negativni brojevi. Problem je riješen tako da se od prvog bajta uzima prvi bit kojim se dojavljuje predznak '+' ili '—', pa se zato zapravo može raditi s brojevima od —32 766 do +32 766.

S brojevima 2 i 3 neće, dakle, biti nekih poteškoća, pa niti s brojem 1234, ali je 10078489 56472 u posve drugačijem položaju. On se jedino može pohraniti u obliku stringa čija će svaka cifra biti predstavljena bajtovima ASCII kóda. No to je rasipanje prostora, jer se jedim bajtom može prikazati do 255 brojeva, a nama treba prostora za samo 10 brojaka u rasponu od 0—9, pa na taj način iskorištavamo samo jednu dvadesetpetinu kapaciteta jednoga bajta.

Rješenje koje se pojavilo nakon mnogo godina mučnog programiranja u strojnomu kodu, nazvano je aritmetika 'plivajuće točke' (floating point). Po ovoj shemi brojevi se prikazuju u obliku osnove i eksponenta. 2345 biti će pohranjen kao 2 345 E3. Svaki od dva dijela od kojih se sastoji ovej broj ima u memoriji fiksni broj bajtova (čita se kao $2\,345 \times 10^3$), pa se tako broj s pomičnom decimalnom točkom sastoji zapravo od cijelih brojeva s pomičnom decimalnom točkom koja može doplivati na bilo koje mjesto. Na primjer, brojevi 234 5678, 23 45678 i 0002345678 u osnovi su isti brojevi s decimalnom točkom na različitim eksponentom određenim, mjestima. Taj način olakšava rad u višim programskim jezicima, jer je količina potrebne memorije za svaki pojedini broj unaprijed poznata, što ne bi bilo moguće kad bi se brojka prikazivale u obliku stringova.

Microsoft BASIC može prikazati brojeve na tri načina, od kojih svaki ima veći stupanj preciznosti: cijeli broj (integer) prikazuje se s dva bajta, što pokriva raspon od —32766 do +32766; jednostruka preciznost (single precision) u kojoj se broj pamti sa 4 bajta, što dozvoljava brojeve od 7 decimalnih cifara; dvostruka preciznost (double precision) od 8 bajtova, što dozvoljava 14 decimalnih brojaka. Korisnik mora odabrati pravu varijantu, jer na taj način čuva memorijski prostor za velike programe i ubrzava izvođenje. Trebe znati da se program koji radi s cijelim brojevima odvija oko deset puta brže nego onaj koji radi s plivajućom točkom, odnosno s dvostrukom preciznošću.

Programiranje u strojnom kódu s plivajućom točkom zaista je vrlo teško, ali na sreću suvremenom je programeru dostupan gotov program u nekom od viših programskih jezika.

Ima različitih trikova ili — da ih nazovemo podobnijim imenom — metoda kojima se služe profesionalni programeri da bi ubrzali i olakšali programiranje. Početniku mogu koristiti, jer ih može primijeniti u jezicima kojima će se i on služiti.

Zastavice

Već smo na strani 80 vidjeli kako se zastavice mogu koristiti pri programiranju u strojnom kodu, ali ne smijemo zaboraviti da su vrlo upotrebljive i u višim programskim jezicima.

Često treba upozoravati na neke opće uvjete u samom pristupu nekom programu. Na primjer, može se dogoditi da obrađujete problem koji se može ticati muških i ženskih osoba. Tada je najbolje na početku programa postaviti zastavicu 'ON' odnosno 'ONA', pa će svi potprogrami naj-

prije testirati zastavicu i prama tome prilagođivati dalji tok programa. Isto tako se, u slučaju rada na programskom paketu koji se istovremeno radi na viša jezika (multilingvalno) može pomoću 'jezičnih' zastavica prelaziti s jednog jezika na drugi

Pohrana podataka

Nije suvišno još jednom naglasiti sve ono što kompjutor *može* primiti u sebe je mnoštvo jedinica i nula. Istina, teško je na prvi pogled odgonetnuti njihovo značenje, jer nitko ne može pouzdano reći da li jedan niz nula i jedinica predstavlja nešto iz igre 'kružić-križić' ili se radi o nekom podatku iz programa za kuru mrsavljenja. Svaki je programer suočen s problemom prevođenja podataka iz vanjskoga svijeta u beskrajne nizove bitova i bajtova.

Najbliži pristup kompjutoru odvija se preko tastature. Na stranicama 12 i 13 vidjeli smo kako se pritisak tipke pretvara u bajtove ASCII kodova. To dobro funkcionira, ali što je sa svakodnevnim informacijama kao što su riječi ili brojevi?

U kompjutorskom, kao uostalom i u svakodnevnom jeziku, riječ je sastavljena od niza slova, od samoglasnika i suglasnika od kojih se formiraju slogovi, pa nema ozbiljna razloga da se riječ ne bi mogla kodirati bilo kojom vrsti znakova ili bajtova. (Treba zapamtiti da ASCII kodovi imaju nulu kao najznačajniji bit, a kada se tipkom SHIFT uključi garnitura 'grafičkih' znakova — ako ih kompjutor ima, onda jedinica postaje najznačajniji bit.)*

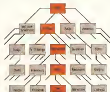
To znači da u memoriju možemo pohraniti najobičniju riječ, recimo apple (jabuka) ili broj nekog doknadnog dijela, na primjer '34thg89/45' kao niz ASCII bajtova, s jednim bajtom ispred niza čija se vrijednost interpretira kao duljina cijeloga stringa: 5apple934thg89/4

Program koji treba da pročita takav string najprije pogleda duljinu stringa, a onda pročita samo onoliko bajtova koliko je početnim bajtom određeno. Budući da se duljina stringa određuje samo jednim bajtom, on ne može biti duži od 255 znakova, jer je 255 (ili heksadecimalno FF) najveća vrijednost koja se može izraziti jednim bajtom.

U savršenijim izvedbama bajt koji označava duljinu riječi (ili duljinski bajt) može biti pohranjen u nekom drugom dijelu memorije s pointerom (pointer je samo adresa početka stringa) upravljenim prema tekstu koji se odnosi. Ako želite promijeniti 'jabuku' u 'naranču', treba samo da okrenete pointer od adrese jabuke na adresu naranče i da promijenite duljinski bajt od 6 na 7 (slova).

Nakon nekoliko promjena još uvijek sve radi sasvim dobro, samo se memorijski prostor koji sadrži stringove (string space) sve više ispunjava odbačenim voćem. To je sličan problem kao kod diska opisan na str. 44-45, ali se rješava na drugačiji način. Posebni servisni program 'Garbage collector' (sakupljač smeća) stupa na scenu kad u memoriji više nema slobodna prostora za neko novo voće. On trči po prostoru za stringove u potrazi za mjestima na koja sa liste imena stringova ne upućuje niti jedan pointer, pa kada nađe takvo mjesto, pomiče aktivne stringove, preokreće pointere i stvara mjesto za nove stringove.

Pogledajmo sada kako sastaviti listu voća kojoj treba osigurati pristup ne kao samostalnim stringovima, već na smisleniji način. Može se raditi o jabukama, breskvama, dunjama, narančama i avokadu... Moža nam trebati da neko voće promijenimo ili neko drugo dodamo ili izbrišemo iz sredine liste. Najjednostavnije je da svakome dodijelimo po dva dodatna bajta koji

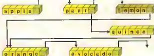


Primjer kako se u kompjutor može pohraniti veliki popis zemalja i gradova: svaka stavka (osim prve 'world' — svijet) upućuje na nekoliko slijedećih. To olakšava postupnost: svi jel. Evropa, Njemačka, Bonn, Bad Honnef (mjesto kraj Bonna), ali istodobno otežava organizaciju programa, jer se nakon svake stavke progresivno povećava broj pointera.

ce pokazivati na ono koje slijedi. Na taj način napravili smo tzv. povezanu listu (linked list), pa ako bismo sada htjeli promijeniti dunju u limun, treba nam samo bajt pointera koji pokazuje na dunju promijeniti tako da pokazuje prema limunu, a onaj limuna usmjeriti prema naranči. To se zorno vidi na slici dolje desno.

Možda bi netko htio napraviti još ambiciozniju

Dolje: jedan od načina za pohranu jednostavnijih lista podataka je tzv. vezana lista. Svaka stavka ima po jedan bajt više koji služi kao veza sa slijedećom stavkom. Da bi se promijenila jedna stavka, na primjer od 'lemon' u 'quince', treba promijeniti bajt na kraju riječi 'pear' tako da upućuje na 'quince'. Sada više ništa ne upućuje na 'lemon', pa će zato to voće nestati sa popisa.



* Ovdje se radi o pojmovima koji se označavaju kraticom MSB (Most Significant Bit — najvažniji bit) i LSB (Least Significant Bit — najmanje važan bit). MSB je krajnji lijevi bit u bajtu i nije ništa više značajan od ostalih bitova, pa i od LSB bita koji se nalazi na krajnjem desnom kraju bajta. Ti bitovi imaju važne funkcije, pogotovo kad se radi o numeričkim podacima ili u slučaju razlikovanja ASCII kodova slova od grafičkih znakova, kao što je navedeno u tekstu. Op. prev.

Bolja sheme od one na prošloj stranici je tzv. stablo: svaka slavka upućuje samo na dvije slijedeće. U računalu (slike dolje) na kraju svake slavke nalazi se dodatna adresa od 2 bajta na koju se upućuje slijedeća slavka. Na početku svake ima također 2 bajta u koje je ubilježena adresa sa koje je stigla uputa. Na taj način lista se može pretraživati i odozdo prema gore, pa je, na primjer, jednostavno naći u kojoj se zemlji nalazi Bad Honnef.

listu podataka sa ograncima i strukturom stabla. Na primjer, nekom može zatrebati lista kontinenata, zemalja i gradova (slika dolje). Svaka veza između podataka zove se „čvor", a pointer je takvog čvora usmjeren prema slijedećem čvoru. Počinje se sa čvorom svijeta koji ima pointere prema kontinentima svijeta, b, c, d ... pointer a vodi u Aziju e, f, g, h ... e vodi u Kinu j, k, l ... j vodi do Delhija.

Unošenje podataka je relativno jednostavno. Treba samo napisati, recimo, „Kina j, k, l" — j, k, l su pointeri do Kine, bez obzira na to gdje se ova zemlja nalazi u memoriji. Ali programiranje ovakve sheme nije lako, jer broj pointera ovisi o broju gradova u svakoj zemlji i o broju zemalja na svakom kontinentu. Jednostavnije je izvesti listu pomoću čvorova samo sa dva ogranka.

U tom slučaju su čvorovi i njihovi pointeri standardnog formata: Kina a, b. Nedostatak je ovakve izvedbe što se svi kontinenti ne nalaze na istoj razini i što se ne može unositi mnogo informacija, jer se na taj način broj čvorova prekomjerno množi. Cijelo stablo visoko deset čvorova sadržavat će 1024 podatka, a ono od dvadeset čvorova — milijun. To je dobro ako želite samo znati koja mjesta postoje u Aziji, ali ako biste htjeli ići natraške i saznati u kojoj se zemlji nalazi Bad Honnef (u Zapadnoj Njemačkoj, blizu Bonna), pred računalom je tada prilično teška zadaća: program mora lutati po memoriji i stalno ispitivati ima li koji pointer što pokazuje na Bad Honnef? To može trajati vrlo dugo.

Najsretnije rješenje je postavljanje pointera ispred i iza svakog podatka. Dakako, ta se lista može napraviti o bilo kojoj temi: može sadržavati podatke o materijalnoj proizvodnji, kompanijama u industriji ili specijalističke podatke za medicinu (vidi str. 86); pitanje u tom slučaju može glasiti: ako je pacijenta ugrizao bijesni pas, u vrućici je, s pjenom na ustima, onda ...

Sortiranje

Tvrdi se da je u ovom trenutku 20 % svih kompjutora u svijetu zaposleno sortiranjem podataka. Iako je ta tvrdnja nastala prije nego što je Sinclair ZX81 počeo svoj pohod po svijetu, ipak vjerno odražava iracionalnu ljudsku strast da podatke sortira po nekom redu. Neki tvrde da je to zato što podaci svrstani po abecednom redu djeluju uvjerljivije, pa čak i onda kad ih prate pogrešni numerički podaci.

Sortiranje fajlova vrši se najčešće po alfabetskom redu. Ako se radi o namještenicima, to će izgledati ovako:

Abble, Tom
Baker, John
Baker, William

Iz popisa se vidi da u jednakih prezimena red ovisi o imenu, ali može ovisiti i o njihovim zaradama:

Lucas, Andrew, 345 26
Baker, William, 122 87
Baker, John, 45 20

Stvari se mogu sortirati na najrazličitije načine. Na primjer, britanska pošta daje popust za veliki broj pošiljaka ako su poštanski brojevi na njima sortirani, što može obaviti posebni program.

Za sada oslonimo kod abecednog reda. Kao primjer uzmimo izvjestan broj fajlova (vidi str. 44-45) koje želimo poslagati po abecednom redu prve riječi. Postoji mnogo načina da se to napravi. Teoretičar Wilhelm Knuth napisao je doduše o toj temi 279 gustih stranica, definirajući sortiranje kao ubacivanje, izmjenjivanje, selekciju, miješanje i distribuciju, no mi ćemo biti skromniji i zadovoljiti se s dvjema karakteristikama što ih program za sortiranje mora poštovati: mora zauzimati što manje memorijskog prostora i obaviti posao u što kraćem roku. Naime, najveći dio vremena program troši na uspoređivanje i odlučivanje (da li treba jedno ime doći ispred ili iza drugog imena), pa zato treba pripremiti tako pametne potprograme da bude što manje komparanja.

Da bismo dobili kakav-takav uvid u problem, napišimo mali program u BASICu koji će sortirati riječi po abecednom redu. Riječi treba unijeti u memoriju pomoću string matrice W$ (Na-

redba REM od remember — sjeti se u BASICu znači da tekst koji slijedi ima funkciju objašnjenja ili upozorenja programeru)

```
20  INPUT"Enter a word and hit Return —
    Return alone to quit  W$(N)
30  IF W$(N)        THEN 50 REM EXIT IF NO
    WORD
40  N  N  1 GOTO 20 REM N COUNTS
    WORDS
50  PRINT N  words entered — sorting
60  SWP=0 REM FLAG TO SHOW IF
    ANOTHER PASS NEEDED
70  FOR K  0 TO N-1
80  IF W$(K)>W$(K+1) THEN SWAP
    W$(K) W$(K+1) SWP  1 REM SEE TEXT
90  NEXT K
100 IF SWP  1 THEN PSS  PSS+1 ? Pass
    PSS  GOTO 60 REM COUNT PASSES  DO
    ANOTHER IF ANY SWAPS ON LAST
110 REM NOW PRINT THE RESULTS
120 FOR K=0 TO N
130 PRINT W$(K)
140 NEXT K
```

Ovaj mali program najprije traži da unesete riječi koje treba sortirati Može ih se dakako unijeti i sa diska ili pomocu DATA naredbe U liniji 80 uspoređuje se svaka riječ — W$(K) — sa slijedecom — W$(K+1) — da se ustanovi koja je veca (BASIC dozvoljava simbol > i sa stringovima) Tako se uspoređuje ASCII vrijednost svakog pojedinog slova u obje riječi Ako je jedna od njih veća test se nastavlja

Pretpostavimo da uspoređujemo riječ banana' s riječi banalnost obje su jednake sve do drugog slova n u banani ASCII kód za n je 110 a za l 108 pa je tako test završen i treba banalnost staviti ispred banana što se postiže naredbom SWAP Ako vas BASIC nema te naredbe treba dodati slijedeći potprogram

```
1000 N$=W$(K)
1010 W$(K)=W$(K+1)
1020 W$(K+1)=N$
1030 RETURN
```

Također treba izvršiti promjenu u liniji 80

```
80 IF W$(K)>W$(K+1)THEN
GOSUB 1000 SWP=1
```

Treba znati da sva velika slova imaju niže ASCII kodove nego mala (vidi str 12-13) pa sve riječi moraju biti pisane jednakim slovima BANALNOST i 'BANANA ili 'banalnost i banana' Program bi se mogao još više usavršiti tako da prije komparanja konvertira sva slova u velika ili mala

Taj način naziva se sortiranje s mjehuricima (bubble sort) jer pri izvođenju lakše riječi plivaju na lijevo To je najlakši nacin programiranja ali kao što cete vidjeti kad program pustite da radi vrlo je polagan jer svaki put treba učiniti N koraka kroz listu i učiniti N komparacija pa je vrijeme potrebno za to srazmjerno kvadratu broja elemenata Knut kaže da u ovom načinu sortiranja nema ničega sto bi se moglo preporučiti osim dopadljiva imena

Vrijeme za sortiranje koje je potrebno dobrom programu bit će proporcionalno N•ln(N) što to u praksi znači vidi se iz ove usporedbe dvaju kriterija

Elemenata	N^2	N•ln(N)
10	100	23
100	10 000	460
1 000	1 000 000	6 908
10 000	100 000 000	92 103

Dobro napisan program sortiranje treba za sortiranje 10 000 elemenata isto toliko vremena kao bubble sort za 100



Tehnika sjeckanog mesa

Ima još jedna zanimljiva tehnika kojom se služe veliki kompjutori Ime joj podsjeca na dosta brutalan postupak što nije slučajno Postupak se osniva na tome da se riječi imena ili svote novca koje treba sortirati zamijene neponovljivim indeksnim brojevima Da bude jasnije o kakvom se postupku radi pretpostavimo da treba sastaviti popis računa različitih poduzeca Treba ih na neki način poslagati u fascikle kako bismo ih u svakom trenutku opet mogli naci Jedan od mogucih načina je da ASCII vrijednost prvog slova elementa zbrojimo s ASCII vrijednošcu sljedećeg slova i podijelimo sa 10 zbrojimo tu sumu s trecim slovom pa podijelimo sa 100 itd

Ime	indeks
john	119 25
fred	115 51
harrods	115 966

Tako je svakom elementu dodan broj koji je garantirano jedinstven a ujedno zauzima malo prostora Osim ovog ima još tisucu drugih sličnih načina na koje se može obaviti sortiranje Sjetimo se da je dio užitka što nam ih pruza kompjutor u izmišljanju postupka koji ce dati jednake rezultate uz što manji utrošak memorijskog prostora

STRUČNI SUSTAVI

Prve slike gore Postoji mnoštvo rutinskih zada taka koji se mogu rješavati samo ako se sakupljene informacije podvrgnu temeljitoj obradi Za primjer neka posluži ova geološka pitalica ako iznad sloja granita ima škriljevca a u površinskim vodama veliki sadržaj magnezija i tada mora pod zemljom biti i zlata Na slici vidimo kompjutorom generiranu sliku područja s mogućim nalazištima označenim crvenom bojom

Druge slike gore Klasifikacije i brojenje čestica u uzorcima krvi lijekova minerala itd jedan je od najdosadnijih ali i vrlo važnih laboratorijskih zadataka Televizijske kamera montirane na mikroskopu daje jednostavnu dvodimenzionalnu sliku a kompjutor broji čestice prepoznavajući ih prema obliku i veličini

Početniku se obično čini da je za korisnika najteži zadatak pisanje programa i savladavanje kompjutorskih jezika no mnogo je veći problem opseg novih pojmova što ga nameću kompjutori i sile ga da se njima služi Na stranicama 94-97 opisana su na primjer vezane pretraživanja banke podataka pri čemu se javljaju teškoće koje ne postoje u svijetu papira i olovke Učeci programiranje korisnik nije ni svjestan da uči i neke posve nove stvari

Onome koji ulazi u svijet kompjutora nije toliko potrebna poduka u programiranju već više poduka u mišljenju To je doduše grubo rečeno ali posebna vrst stručnih programa nudi upravo takvu poduku i pomoć To je softver osnovan na saznanju i spada u grupu tzv stručnih sustava (Expert systems)

Najjednostavniji među tim programima je 'menadžer banke podataka' (vidi str 94-97) koji registrira i daje različite vrsti informacija 'Nešto je prodano tvrtki Binks & Co 14 siječnja 1982 Što je to bilo? Program koji samo odgovara na takva pitanja nije u pravom smislu dovoljno stručan premda može raspolagati velikom količinom informacija Na višoj razini su oni programi što su ih sastavili vrhunski stručnjaci koji mogu registrirati još i mnoštvo dodatnih informacija Dobrim dijelom bave se i nesigurnim procjenama korisnika koje na inteligentan način uzimaju u obzir Pogledajmo na primjer kako se od banke podataka može dobiti savjet o popravku automobila Konverzacija se može voditi ovako (A je korisnik a B je stručni sustav)

A Neće upaliti
B Ima li benzina u spremniku?
A Ima
B Nije li akumulator ispražnjen?
A Ne bih rekao
B Kolika je vjerojatnost?
A 80 %
B Treba prekontrolirati priključke

Kao što se vidi program prihvaća čak i sasvim nesigurne odgovore Ispitivanje se može protegnuti i na svjećice a korisnik može izjaviti da je samo 50 % siguran u njihovu ispravnost Na koncu će program navesti niz mogućih grešaka s postotkom vjerojatnosti uzroka kvara

Za sastavljenje takvog programa treba angažirati stručnjaka za određeno područje ali je nezgoda u tome što se njima ne služe samo tehnički obrazovani pojedinci već i oni koji o mehanici imaju vrlo mutne pojmove te su im i sudovi vrlo nesigurni Stoga nije lako sastaviti program koji će zaključiti da je greška najvjerojatnije u labavom kablu na kapi razdjelnika ako je akumulator 80 % ispravan u spremniku ima benzina a svjećice su 50 % u redu Međutim tvrdi se da ima stručnjaka koji će uzevši u obzir sve okolnosti ipak sastaviti uspješan program ove vrsti

Još bi bio bolji program koji bi sam učio Trebalo bi ga opskrbiti činjenicama i podacima koje bi dobijao s diska a onda dedukcijom iz općih sudova zaključivao pojedinačne

Zamislite da ste vlasnik nekog poduzeća pa naredite svom kompjutoru da pregleda stanje skladišta i plaće zaposlenih unatrag dvije godine Program će poslušno zaroniti među fajlove i nakon nekog vremena izbaciti zaključak čini se da svaki put kad u zimskim mjesecima u skladištu ponestane zelene tkanine upošljavate dva nova radnika Nakon toga ne preostaje vam ništa drugo nego da se zahvalite i da svake godine odmah nakon ljetnog odmora naručite veću količinu zelene tkanine i tako uštedite na plaćama

Richard Forsyth s Tehničkog fakulteta u Londonu sastavio je sličan program koji je opskrbio bolničkim podacima o srčanim bolesnicima Uzeo je u obzir sve što se o njima znalo a program bi onda dao najpouzdanije indikatore ozdravljenja Tako je ustanovljeno da će pacijent vjerojatno preživjeti ako njegov prosječni arterijski tlak (u mm stupca žive) minus izlučivanje urina (u ml/h) iznosi više od 61 Ali ako je ta cifra manja — nade nema To je iznenadilo liječnike Premda su godinama promatrali srčane bolesnike nisu bili u stanju tako precizno uočiti vezu između tih dviju indikacija kao što je to učinio kompjutor

U osnovi takvi bi se programski sustavi mogli nazvati generatorima programa' Za sastavljanje nekog knjigovodstvenog programa neće trebati istraživati sve tajne knjigovodstva niti način na koji će kompjutor ući u njih on će dobiti sve knjigovodstvene podatke unatrag nekoliko godina sam će stvoriti zaključke i ustanoviti pravila Možda će trebati još nekih dopuna i objašnjenja ali će generator programa ipak obaviti veći dio intelektualnoga posla

Takav program se može upotrijebiti i u slučajevima kad se ne znaju pravila igre Pretpostavimo da neki ekolog proučava ponašanje lisica što su se naselile po gradovima, pa će zbog toga podijeliti gradsko područje na kvadrate od 100 metara i ubacili u kompjutor podatke o svim lisicama u svakom kvadratu i sve što pretpostavlja da bi moglo utjecati na njihovo ponašanje Kada to kompjutor prouči može izbaciti zaključak mužjaci za svoja staništa najradije odabiru garaže pogotovo ako se u krugu od 100 metara nalazi pekara Ženke više vole ružine grmove knjižare i trgovine sešírima

Takvi se programski paketi počinju proizvoditi u laboratorijama za umjetnu inteligenciju Jedan od njih pod imenom 'analogno konceptni sustav učenja' (Analog Concept Learning System) nastao je na edinburškom Sveučilištu i može se vrtiti i na malim računalima poput Apple-a

ZIPFOV ZAKON

U radu s kompjutorima je najveći problem, ali i najveći izazov (i najveće zadovoljstvo) činjenica da se mogu znatno brže obraditi mnogo veće količine informacija nego što bi se ikada moglo s papirom i olovkom. Podaci se mogu obrađivati na mnogo različitih načina i često smo iznenađeni novim mogućnostima što se pred nama otvaraju.

Američki sociolog George Zipf* otkrio je negdje četrdesetih godina zanimljive zakonitosti koje se pojavljuju kad se radi o velikom broju informacija. Iako je Zipfov rad vrlo značajan upravo za kompjutorske znanosti, on je cijeli taj ogroman posao obavio služeći se olovkom i papirom. Počeo je proučavati učestalost riječi u engleskom jeziku brojeći ponavljanje svake pojedine riječi u djelima velikih proznih pisaca. Izradio je listu s najčešće ponavljanom riječi, za njom drugu po rangu i onda redom sve ostale. Izradio je i grafički prikaz rezultata svog istraživanja.

Grafikon je bio upravo onakav kakav se mogao i očekivati: što je riječ neobičnija, manje se upotrebljava. Ali krivulja grafikona nikada ne dotiče apscisu, jer se uvijek iznova pojavljuju neke nove rijetke riječi i produžavaju krivulju sve dalje i dalje.

Onda je Zipf nacrtao još jedan grafikon, ali sada kao logaritam učestalosti u zavisnosti od pojavljivanja riječi na tablici. I, što je bilo još zanimljivije, dobio je umjesto krivulje — pravac. Onima što nemaju pojma o matematici čini se to možda nevažnim, ali time se red i učestalost pojavljivanja riječi može izraziti jednadžbom

$$\log F = -k(\log R) + 1$$

gdje su k i 1 konstante. Ali 1 se može prikazati kao logaritam neke druge konstante, na primjer m, pa jednadžba glasi

$$\log F = -k(\log R) + \log m$$

ili

$$\log F = \log((m/R) + (k))$$

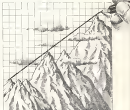
Pokazalo se da je k blizu jedinici, pa kad se antilogaritmira, to izgleda ovako

$$F = m/R$$

što, jednostavno rečeno, znači da je učestalost riječi obrnuto proporcionalna njihovu mjestu na listi. Na primjer, treća riječ na listi javljat će se u tekstu tri puta rjeđe nego najčešća riječ. Stota po redu javljat će se stotinu puta rjeđe.

Preostalo je još samo da se utvrdi koliko često se javlja najčešća riječ. Zbog toga ne treba trošiti tjedne brojeći riječi u Snu ljetne noći, već jednostavno zbrojiti 1/1 + 1/2 + 1/3 + ... 1/n. Premda se zbroj u početku naglo povećava, kasnije teži izjednačenju. Može se napisati mali program koji će to izračunati:

* G. K. Zipf: Human Behavior and the Principle of Least Effort, Addison-Wesley, 1949.



```
10 K=1 N=0
20 N=N+1/K
30 PRINT K N
40 K=K+1
50 GOTO 20
```

Kompjutor će dati sljedeće rezultate:

K	N
10	2 929
100	5 187
1000	7 485
10000	9 788
100000	12 091

Kako se k povećava, n raste sve sporije i kada postane vrlo velik ustalit će se oko 12. Može se, dakako, nastaviti s programom sve do k=milijun, deset milijuna itd., ali se zato troši izuzetno mnogo vremena, a nema nekih novih značajnijih spoznanja.

Zipf je također ustanovio da posvuda vlada ista ili vrlo slična zakonitost. Primijenio je svoj zakon na veličinu gradova i mjesta: u bilo kojoj zemlji drugi grad po veličini po prilici je dva puta manji od najvećega, treći je tri puta manji, pa sve tako do najmanjih sela. Prema tome, ako se zna broj stanovnika neke zemlje, može se, recimo, izračunati koliko ima gradova koji imaju između 100 i 200 tisuća stanovnika.

Vrlo zanimljive rezultate dalo je istraživanje poduzeto u velikim američkim gradovima. Pokazalo se da je bračnu sreću u susjednoj četvrti našlo upola manje ljudi od onih koji su se vjenčali u četvrti u kojoj stanuju. Za trećinu manje ih se potrudilo da nađe supružnika tri bloka dalje itd.

Zipfov zakon se može primijeniti i na poslovanje. Ako znate ukupnu sumu za koju su posljednje godine prodani mikrokompjutori, možete se poslužiti Zipfovim zakonom da biste po prilici ustanovili promet različitih kompanija. Prva kompanija po veličini prodala je po prilici dvostruko više od druge po veličini itd.

MODELIRANJE

Jedan od najkonsnijih poslova što ih kompjuton mogu obaviti je modeliranje situacija koje se još nisu dogodile. Kompjutorski modeli konste se za predviđanje ishoda znanstvenih pokusa, trgovačkih poslova, vođenja ekonomije, ratova i djelovanja novih oružja. Također se konste za pripremu meteoroloških prognoza, predviđanje kretanja bankovnih kamata i prirasta stanovništva tokom slijedećih pet godina.

Matematički modeli dakako nisu novost, preve se već stoljećima. Pomoću Newtonovih zakona gibanja mogu se lako stvoriti matematički modeli kretanja svemirskih sustava, kao što su sunce, zemlja i mjesec, koji pretvoreni u niz jednadžbi daju položaje bilo kojeg od tih tijela u bilo kojem času. Na taj način astronomi predviđaju pomrčine sunca ili mjeseca mnogo godina unaprijed ili, što je još korisnije, sastavljaju navigatorske tablice za plovidbu i letenje.

Na žalost malo je pojava u životu koje se u potpunosti daju predstaviti dovoljno jednostavnim jednadžbama koje se mogu rješavati olovkom i papirom. Svatko tko poznaje osnove mehanike može riješiti jednadžbu gibanja topovske granate u vakuumu, ali teško je riješiti jednadžbu gibanja granate u atmosferi, jer atmosfera usporava granatu proporcionalno kvadratu njezine brzine.

Problem se najlakše može riješiti korištenjem kompjutorskog modela. Put granate treba podijeliti u mnogo malih odsječaka vrlo kratkoga trajanja, recimo veličine tisućinke sekunde. Počinje se promatranjem granate na početku prvog vremenskog intervala. Poznata je njezina brzina, pa se može izračunati koliko će je usporiti otpor zraka za vrijeme tog intervala. Iz prosječne brzine u svakom intervalu izračunava se vertikalna i horizontalna komponenta gibanja granate, a rezultat se dodaje zbroju rezultata ranijih intervala. Za slijedeći interval određuje se nova brzina i položaj granate. Proračun se izvodi pomoću ovog malog programa



```
 50 U  W COS(AN) V  W*SIN(AN)
100 FOR T  0 TO 1000 STEP DT
105 PRINT X Y
110 X  X+U*DT Y  Y+V*DT
116 IF Y<0 THEN STOP
120 Z  U↑2+V↑2
130 W  W  K*DT*Z
140 U  W*U Z↑ 5
150 V  W*V Z↑ 5−32*DT
160 NEXT T
```

```
10 K  01 DT = 1
20 INPUT Muzzle velocity W
30 INPUT Angle of elevation AN
40 AN−AN*3 14 2 90
```

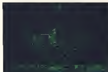
U liniji 10 postavljaju se dvije konstante: K za otpor zraka koji ovisi o obliku i težini granate i DT, vremenski interval koji u ovom slučaju iznosi 0.1 sekunde. Linija 20 traži upis brzine u stopama na sekundu, a linija 30 kut elevacije. U slijedećoj liniji (40) stupnjevi se pretvaraju u radijane, jer BASIC računa isključivo u radijanima. U liniji 50 izračunava se horizontalna (U) i vertikalna (V) komponenta brzine.

U liniji 100 počinje petlja koja izračunava let u pojedinim intervalima (ako 1000 nije dovoljno za cijelu putanju, broj intervala se može povećati), a linija 105 ispisuje položaj granate po horizontalnoj (X) i vertikalnoj (Y) osi. U liniji 110 izračunava se slijedeći položaj pribrajajući rezultatu udaljenost koju granata prede horizontalno (U•DT) po X osi i vertikalno (V•DT) po Y osi.

V će postati negativan kad granata prede tjeme putanje i počne padati prema zemlji. Linija 115 ispituje da li je granata dotakla zemlju, pa ako jest, zaustavlja program. U liniji 120 izračunava se kvadrat brzine granate, što je prema Pitagorinu poučku suma kvadrata horizontalne i vertikalne komponente brzina. U liniji 130 izračunava se promjena brzine W do koje dolazi zbog otpora zraka.

Linija 140 izracunava novu horizontalnu brzinu U, a u liniji 150 ponavlja se isto za V, ali s faktorom (32•DT) kako bi se uzela u obzir i akceleracija prema dolje prouzročena gravitacijom. U liniji 150 izračunava se nova vertikalna brzina uzimajući u obzir otpor zraka i gravitaciju.

Program na koncu ispisuje numeričku listu X i Y položaja granate, ali ne bi bilo teško da se rezultat prikaže i grafički, na ekranu ili pomoću printera. Ipak ne smije se prešutjeti da su ti proračuni prilično netočni i to upravo zbog metode malih koraka — intervala. Sasvim mala pogreška svakim novim korakom postaje sve veće, pa je cijeli proračun (u samo nekoliko koraka) suviše jednostavan da bi se njime prikazalo ono što se u stvarnosti događa. Ako su intervali dovoljno kratki, mogu se zanemariti komplikacije oko uzajamnog djelovanja gravitacije i otpora



zraka jer to otežava rješavanje jednadžbi. Ali i pored svega, taj je račun zanimljiv već i zbog toga jer je bio jedan od prvih zadataka što su ga rješavala elektronička računala još tokom prošlog svjetskog rata.



Solarno jedrenje

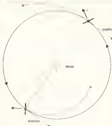
Zanimljiv predmet za stvaranje kompjutorskog modela bile bi solarne jedrilica, poput one iz filma Tron, kojom junaci filma putuju od planete do planete. Filmaši se očito nisu mnogo trudili da princip njena kretanja riješe na uvjerljiv način, jer su je prikazali kako jednostavno lebdi nad zemljom gonjena zrakom sunčeve svjetla.

Zamišljeno je da se svemirski brod kreće pomoću ogromnog jedra od vrlo laganog reflektirajućeg materijala. Upravlja se okretanjem jedra pod različitim kutovima prema zrakama sunca. Jedro ima zrcalnu površinu od nekoliko kvadratnih kilometara, a tlak sunčeve radijacije (čak i fotoni imaju moment sile gibanja) proizvodi silu koja tjera brod. Sila može ubrzati ili usporiti brod i tako ga približiti ili udaljiti od sunca.

Ako se jedro postavi pod takvim kutom da brod ubrzava, on će se, savladavajući gravitaciju, udaljavati od sunca. Drugačijim postavljanjem jedra brzina će se smanjivati i brod će se vraćati prema suncu. Budući da su gravitacija i tlak sunčevih zraka obrnuto proporcionalni kvadratu udaljenosti broda od sunca, može se uspješno jedriti bilo gdje u sunčevu sistemu.

Modeli korak po korak, poput ovoga, koriste se još i onda kad su proračuni previše složeni da bi se obuhvatili odjednom. Za to je dobar primjer meteorološke prognoze. Meteorolog zna (ili barem misli da zna) fizikalne zakone po kojima se ponašaju atmosferske zbivanje. Prema izvještajima što ih prima od meteoroloških postaja iz svih krajeva svijeta, računanjem korak po korak meteorolog može predvidjeti vrijeme za nekoliko sati unaprijed. Međutim, da bi se potrebni proračuni završili dovoljno brzo, moraju se raditi na najmoćnijim postojećim kompjutorima.

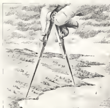
Ekonomija je poput meteorologije jednako složena i nepredvidiva, pa se i za predviđanje ekonomskih kretanja koriste kompjutorski modeli (sastavljeni od mnogo malih jednostavnih jednadžbi ekonomskog ponašanja). Na žalost, niti jedan od modela nije naročito uspješan. Ekonomiju ne čine nerazumna tijela koja se slijepo pokoravaju fizikalnim zakonima, već žive i razumne biće koja neprestance nastoje shvatiti ekonomska događanje. Na taj način se izračunavanje komplicira, te još nitko nije došao do pravog rješenja ekonomske jednadžba. Stoga ne iznenađuje što kompjutorski modeli na tom području nisu previše djelotvorni.

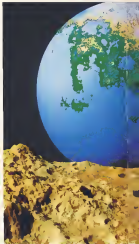


U Disneyevu filmu Tron ima mnogo spektakularnih kompjutorom generiranih grafičkih rješenja. Ovo što ga vidimo na slici predstavlja pokušaj solarnog jedrenja. Šteta je što autori čini se nisu imali pravog pojma kako bi takva naprava uopće mogla funkcionirati.

Lijevo: jedrenje u orbiti pomoću solarne energije: kormilar upravlja brodom naginjući svoje jedro ogledalo tako da ga sila reflektiranog svjetla tjera po planiranoj putanji (gore). Kako centrifugalna sila baca brod sve dalje od sunca, kormilar mora naginjati jedro tako da smanji brzinu (dolje), što ga opet vraća prema suncu. Naginjući jedro naizmjence gore i dolje, kormilar može održavati brod točno u putanji, ali za to mu, u prvom redu, treba veliko strpljenje i dobar kompjutor.

FRAKTALI





Tko je ikad promatrao kompjutorske slike mogao je primijetiti njihov veliki nedostatak: na prvi pogled je vidljivo da su djelo stroja, a ne ljudske ruke. U prirodi su oblici mnogo zapleteniji, nepravilniji i sasvim drukčiji od glatkih i pravilnih oblika što ih dobijamo na ekranu ili printeru.

Srećom, nedavno je i na tom polju postignut veliki napredak zahvaljujući radovima matematičara Benoita Mandelbrota. On je otkrio jednostavan matematički način kojim se može opisati i rekonstruirati svo bogatstvo oblika planinskih vrhunaca, morske obale ili razlistana drveća — pa sve do onog nevjerojatno finog pulsiranja vena i arterija u ljudskom tijelu. Nazvao ga je tehnikom fraktala.

Mandelbrot je počeo svoja istraživanja od onog jednostavnog pitanja što ga postavlja svaki nastavnik i svaki student geografije: koliko je zapravo dugačka obala? Ako pogledamo kartu istočne obale SAD i pokušamo odrediti njenu dužinu služeći se šestarom čije smo krakove otvorili na razmak koji odgovara udaljenosti od 400 milja, ustanovit ćemo, na primjer, da udaljenost od zaljeva Fundy do Key Larga u Floridi iznosi oko 1685 milja. Ponovimo li mjerenje u otsječcima od 100 milja, dužina obale povećat će se na 1750 milja i to na prilično ravnom dijelu obale.

Budemo li se naoružali strpljenjem i na kartama manjeg mjerila nastavili mjerenje u još manjim otsječcima od 50, 25, 10, 5 ili 1 milje pa čak i manje, vidjet ćemo da će se dužina obalne linije dramatski povećavati. Nastavimo li dalje, pa povećamo preciznost na centimeterske odsječke, trebat će mjeriti obujam svake stijene. Smanjujući dalje mjerilo na milimetre i desetinke milimetara, dužina obale eksplozivno raste, jer mjerenje se sada provodi oko milijuna kamenčića. Idući sve dalje, do tisućinke milimetra, mjerimo zapravo zrnca pijeska i hrapavost površine kamenja. Daljim smanjivanjem omjera zaći ćemo u kristaličnu strukturu stijena. Što zapravo pretstavlja pojam obalne linije? Daljim smanjivanjem mjerila ulazimo u jezgru atoma gdje se 'materija' sastoji od gotovo potpuno prazna prostora. Dužina obalne linije — ako taj pojam još uopće ima nekog smisla — iznosi sada milijarde milja.

Razmišljajući na ovakav način moramo zaključiti da obalna linija uopće nije linija, jer kad bi to bila, imala bi neku određenu dužinu. Nakon ovih saznanja obalne linija postaje nešto poput prostora nejasno omeđene trake što teče uzduž obale na geografskoj karti. Znanost tvrdi da pravac ima jednu, površina dvije, a volumen tri dimenzije. Mandelbrot i njegovi sljedbenici smatraju da se obala nalazi negdje po sredini između prve i druge dimenzije.

Za dokaz možemo se poslužiti nečim nalik na rekurzivni postupak. Počinjemo s velikim likom (gore desno) koji se može u nizu ponavljati n puta. Ako ga smanjimo za određeni faktor i svaki pravac pretvorimo u umanjenu sliku samoga lika, pa postupak ponavljamo sve dalje i dalje, uskoro će cijeli papir biti prekriven gustim spletom linija. Iako znamo da je svaka uma-

njena verzija početnog lika samo linija pa je zato jednodimenzionalna vidjet ćemo ako pogledamo iz daleka kompaktnu površinu prekrivenu crtama, pa budući da je to površina, mora da je dvodimenzionalna. Naučani smo da se misleći o dimenzijama služimo cijelim brojevima — prva, druga ili treća dimenzija — ali to, matematički gledano, nije nužno. Broj 1,3 sasvim lijepo se uklapa u teoriju fraktala. Ako vaš kompjutor ima grafiku, neće vam biti teško sastaviti jedan mali program koji će prikazati nekoliko faza tog procesa.

Iako je ovaj lik izlomljen i kvrgav poput obalne linije, ipak je još uvijek previše pravilan da bi odgovarao prirodnim uvjetima. Pravilnost se može izbjeći postupkom slučajnoga smjera nalik onom iz kompjutorske igre Pijani Paja gdje program crta lutanje pijanca što tetura po ekranu u svim smjerovima. Nakon nekog vremena ekran je ispunjen njegovim tragovima — poput fraktala ili razmotanog klupka vune.

Igra „pismo glava" puno je starija od kompjutora i svatko je se sjeća još iz djetinjstva. Marko i Sanja naizmjence bacaju novčić, pa ako padne „glava", novčić je Sanjin, ako na tlu ostane „pismo", novčić je Markov. Kada bismo grafički prikazali kretanje njihovih dobitaka ovisnih o čistoj sreći, slika bi neodoljivo podsjećala na presjek geografskog reljefa.

Mandelbrot je iskoristio tu ideju pa je konstruirao još razvedeniju obalu čija linija nije pravilna ali je ipak globalno usmjerena i nikad ne zakreće tako oštro da bi se preklapala (kao kod Pijanog Paje). Gledano matematički, to znači držati dimenziju obale između 1 i 2. Ista stvar u tri dimenzije daje sliku pejzaža, a uz dodatni programerski napor nastat će slika poput ove nazvane „Izlazak planete iza Prokletog brda" (snimak s nikad neodržanog interplanetarnog leta).

Gore: fraktal se može prikazati kao jednostavni trostrani oblik (gore 1). Ako se svaka stranica zamijeni sa smanjenom verzijom istog oblika (2), pa se dalje nastavi s razlaganjem po istom principu (3 i 4), uskoro će cijela površina papira biti prekrivena nečim što nije ni linija ni ploha, ali ima osobine jedne i druge.

Lijevo: na prvi pogled čini se da je to neki snimak Zemlje načinjen s mjeseca. Ali, kad se pažljivije pogleda, vide se kontinenti otoci i mora kakvi nigdje na ovom svijetu ne postoje. Richard F. Voss iz istraživačkog odjela IBM-a služeći se formulama koje je postavio Benoit Mandelbrot, natjerao je kompjutor da glumi Stvoritelja i da stvori planetu i njezin satelit koji su tako slični, a ipak potpuno različiti od naše rodne Zemlje i njezina blijedog Mjeseca.





Najrasprostranjenije vrste softverskih paketa zapravo imitiraju standardne papirne dokumente te ako uđete u bilo koji ured na svijetu i uzmete komad papira ustanovit ćete da se skoro svaki može svrstati u jednu od grupa od kojih svaka ima svoj kompjutorski ekvivalent Na primjer obrasci i liste odgovaraju programskom paketu Database menager poslovno dopisivanje odgovara programskom paketu 'obrada teksta (Word processor) a knjigovodstveni poslovi programskom paketu tipa Visicalc

Baze podataka

Prvi kompjutori bili su zamišljeni kao strojevi za izvođenje kompliciranih proračuna Mogli su izračunavati na primjer putanje artiljerijskih zrna ili stanje nakon eksplozije atomske bombe ali su pritom baratali s prilično malom količinom numeričkih podataka

Mikrokompjutori su u početku slijedili isti put kako su imeli sasvim primitivne uređaje za memoriranje bili su sposobni raditi prilično složene operacije ali sa relativno jednostavnim podacima Međutim u svakodnevnom životu čovjek želi obavljati relativno jednostavne operacije s velikim brojem podataka Konačno posao uredskog službenika u osnovi nije osobito kompliciran, komplicirano je baratanje s mnoštvom podataka sadržanih u gomilama papira Zbog toga kompjutorsku bitku na tom području nisu dobili kompjutori već jeftine naprave za masovnu i jednostavnu pohranu podataka (opisano na str 176-177)

Međutim samo spremanje podataka nije dovoljno jer treba postojati i neka struktura koja će omogućiti brz i jednostavan pristup pohranjenim podacima Uredski ormarić bio bi besmisleni komad namještaja kad ne bi imao onih ladica i pretinaca u koje se stavljaju dokumenti i sheme po kojoj se kasnije svaka informacija može pronaći Mora postojati indeks i register za praćenje unošenja i iznošenja dokumenata za kontrolu pristupa dokumentima te za njihovu dopunu Svaki kompjuterizirani sustav pohrane informacija mora biti opremljen sličnom strukturom i glavna programerska zadaća svodi se upravo na stvaranje što praktičnijeg i jednostavnijeg načina pohrane i nalaženja informacija

Organizirani arhiv informacija općenito se naziva baza podataka (database) a uza nj postoji još i upravitelj (manager), program koji indeksira i sortira podatke

Baza podataka i upravitelj baze podataka' su termini naslijeđeni od velikih kompjutora i (razloge ćemo kasnije navesti) ne pristaju najsretnije u svijet mikroračunala Sve češće se susreće termin upravitelj informacija' (information manager), a ja pretpostavljam da ćemo uskoro govoriti jednostavno o infobazama' (infobase) Suvremeni sustavi pohrane informacija teže više prema formama koje će biti što pristupačnije korisnicima nego što se bave profinjenim strukturama od kojih se sastoji konvencionalni upravitelj baze podataka

Pristupajući dizajniranju baze podataka treba biti svjestan velikih razlika između fajlova kojima manipulira operativni sustav i konvencionalne informacije što je znamo iz svakodnevne prakse Sjetimo se onoga što radimo u uredu kada bilježimo informaciju o nekom kupcu najprije na indeksnu karticu napišemo ime prezime adresu i telefonski broj a onda obično dodajemo još nekoliko riječi da bismo upotpunili informaciju

Sve se to može pohraniti kao kompletni kompjutorski fajl ali uz dva nedostatka Prvo pohranjeni podaci mogu se kasnije naći samo prema imenu fajla i drugo većina operativnih sustava pridružuje takvom fajlu mnogo slobodna prostora na disku pa je zbog toga ograničen broj fajlova koji se na njemu mogu upisati Nakon nekoliko zabilježenih informacija treba preći na novi disk

Ono za čim se teži je mnogo finija struktura koja se neće ograničiti samo na prezimena složena prema abecednom redu već će omogućiti pronalaženje informacija prema bilo kojem podatku što je u njoj sadržan Na tržištu postoji više takvih programa oni se ponešto razlikuju u detaljima ali u osnovi svi su slični shemi što ćemo je ovdje opisati To je program pod komercijalnim nazivom superfile britanske proizvodnje

Kod superfajla susrećemo se s pojmom zapisa (record) To je skup izvjesnog broja podataka koji treba da se nađu zajedno a sastoji se od jednog ili više stavaka (item) Svaki stavak ima 'etiketu' (tag) koja se uvijek sastoji od jedne riječi i iznosa (value) što sadrži samo informaciju

Evo za primjer jednog personalnog zapisa

IME = TOMA
PRIME = PALČIĆ
ADR = VRTNA KUĆICA
ADR = GRAŠKOVA ULICA
ADR = ZEMLJA ČUDESA
OD = 50

PRIME je kao što se može zaključiti etiketa za prezime ADR za adresu a OD za osobni dohodak Etikete se mogu po volji ponavljati kao što se u ovom primjeru ponavlja ADR a cijeli zapis može sadržavati bilo koji broj informacija Nije potrebno unaprijed specificirati dužinu stavaka niti na početku saopćavati superfajlu koliko će stavaka sadržavati zapis Ako se tokom rada odluči dodati i podatak o bračnom stanju treba samo dodati novu etiketu BRASTA

Na suprotnoj stranici uredski poslovni softver uvodi sav stari uredski pribor kojim se ljudi svakodnevno služe u jedan sasvim novi medij — u kompjutor


GROSS WAGES
A-Z

```
Surname          [Williams                                        ]
Christian Name   [Stephanie                          ]
Company          [Medium Rare Bistro                              ]
Address          [115a Sutton Drive                               ]
                 [Newton Burrows                                  ]
                 [                                                ]
                 [                                                ]
County           [Wiltshire                          ]
Reference Code   [156 ]
Remarks          [A good little meal for the price of a snack     ]
                 [                                                ]
                 [                                                ]
Credit Amount    [250.17] Date last adjusted [13] [ feb ] [1983]   Amount [123.56 ]
                 [250.17]                    [30] [ apr ] [    ]          [123.78 ]
                 [      ]                    [5 ] [ may ] [    ]          [47 34  ]
Total Credit     [500.34  ]                                        Total  [204.68 ]
```

Tipčan kompjutorski zapis Southdela Superfile softvera

1 Ugiato zagrade odvajaju rubrike (u koje se unose i na kojima se iskazuju informacije) od ostalka ekrana gdje su sadržene napomene i objašnjenje baš kao na papirnom uredskom formularu

2 Ako se prezime Williams utipka u rubriku prezime (surname) na ekranu se odmah pojavljuje zapis s podacima koji se tiču osobe imenom Williams i svih drugih osoba istoga prezimena Štoviše možete zatražiti podatke za osobu čije prezime zvuči kao Williams

3 Superfajl softver omogućuje korisniku definiranje izvjesnog broja rubrika kao logičkih ekvivalenata pomoću kojih se može osim po prezimenu isti zapis pronaći prema imenu grada Newton Burrows ako ga se unese u bilo koju od 4 adresne rubrike

4 Svaki zapis ima jedinstveni referentni kôd

5 Kad je potrebno mogu se dodaveti primjedbe prema kojima se također može pronaći zapis U ovom primjeru se zapis može pronaći i prema riječi snack kojom završava primjedba

6 Ova je rubrika tako organizirana da je isključeno slučajno unošenje pogrešnih podataka

7 Ovaj zapis može se pronaći tražeći osobu s kreditom većim od 250$

8 U ovoj rubrici mogu se vršiti i aritmetičke operacije Ovdje vidimo ukupne troškove gospodine Williamsa

Pomoću Database managera mala poduzeća mogu voditi evidenciju svojih stranaka neka garaža može voditi liste s opisima i brojevima rezervnih dijelova mjestima uskladištenja i brojem primjeraka Bolnice mogu voditi evidenciju pacijenata njihovih bolesti podatke o njihovim liječnicima i propisanom tretmanu Tako će odmah biti dostupan svaki dio tražene informacije

Kao i mnogi drugi slični programi i superfajl ima posebne programe (utility) koji omogućuju korisniku da na ekranu prikaže formulare da unosi informacije i da ih kasnije pronalazi eventualno mijenja itd Pored toga ima i generator izvještaja (Report Generator) koji automatski sastavlja tabelirane izvatke iz baze podataka

Dobri programski paketi mogu nacrtati obrazac na ekranu da bude isti kao na papiru te izvršiti sve potrebne obrade Na primjer, kad treba izračunati osobne dohotke namještenika zapisu Tome Palčića dodaje se samo nove etiketa pod šifrom BODSAT pod kojom su spremljeni podaci o normi svakog radnika a program će automatski bilježiti svaki radni sat te će na kraju mjeseca izračunati točan iznos osobnog dohotka i tiskati isplatnice

Iskustvo je pokazalo da se unošenjem mnoštva informacija često prave glupe greske pa se zato vjerodostojnost informacija pažljivo kontrolira Zbog toga postoji mehanizam koji kontrolira da li su brojevi zaista brojevi — da nečija placa nije pogresno utipkana kao XXw' sto bi pri izradi platnih spiskova proizrodilo velikih neprilika To se rjesava tako da se svaki podatak otipka dva puta pa ako se oba poklapaju program ih tek onda prihvaca kao vjerodostojne

Standardni programski paket je tzv generator izvještaja' (Report Generator) koji omogućuje sastavljanje standardnih izvještaja na primjer liste dužnika što su dužni iznad određene svote ili dužnika čije su obaveze predugo nepodmirene ili se u bolnici jednom tjedno mogu praviti izvještaji koji iskazuju zauzete krevete po odjelima zaposlenost osoblja i potrosak medicinskog materijala

Čini se paradoksalno ali kompjutorska obrada podataka često donosi neka nove probleme koji uopce ne postoje u klasičnom poslovanju Na primjer kad bi se izvještaji o iskoristivosti bolničkih kapaciteta pravili na klasičan način, zahtijevali bi nekoliko novih službenika Njihove place i ostali troškovi predstavljali bi znatne izdatke te se sakupljanje takvih informacija uopce ne bi isplatilo Naprotiv u kompjutorskom sustavu troškovi prikupljanja i obrade podataka su tako niski da u tom smislu ne postoje nikakva ograničenja Prirodna posljedica jeftinog prikupljanja i obrade podataka je hiperprodukcija papira koji sadrže toliko mnoštvo podataka da se gubi njihova svrha

Uvođenjem kompjutora u poslovanje teži se ne samo smanjivanju troškova poslovanja nego i uštedi vremena i to ponajviše rukovodećih ljudi u radnoj organizaciji Automatizacijom rutinskih poslova preostaje više vremena za kreativni rad te je za mnoge loše službenike to pravi šok jer postaju svjesni da nestaju poslovi na koje se troši mnogo vremena a malo kreativnog truda

Jednostavni programski paket za manipulaciju bazom podataka može biti vrlo koristan kad treba samo pohraniti ili doznati dio neke informacije Međutim u životu se stvari često kompliciraju U poslovima se često susrećete s istom osobom (vidi Zipfov zakon str 87 o pojmu učestalosti) i treba vam recimo njegovo ime adresa i telefonski broj ali ako ce svaki put praviti isti zapis i ponavljati iste podatke Ako ste prodali deset stvari gosp Smithu i unijeli ga u bazu podataka kao Smith može se dogoditi da jedanaesti put pogriješite i otipkate Smythe umjesto Smith pa ga idući put kompjutor neće naći Osim toga ako Smith promijeni adresu ili neki drugi osobni podatak zgodno je da se promjene unese samo na jedno mjesto

Sve to znači da vaša baza podataka mora biti sposobna da povezuje različite zapise tako da svaki zapis o poslu sa Smithom može biti vezan za zapis koji sadrži Smithovo ime prezime i

Pretpostavimo da ste vlasnik brodarske kompanije i netko vam javi: žena vašeg mornara rodila je dijete. Stanuju na adresi Rose Cottage. Tko je sretnik, na koji brod je ukrcan i gdje se nalazi? Pretraga baze podataka počinje od dobivene informacije.

Potraga počinje od A – ROSE COTTAGE traži se registar TOMA BOWLINEA

TOM
BOWLINE
ROSE COTTAGE
BACK LANE
LYMINGTON

Sljedeći stupanj: PREZIME – BOWLINE traži se registar posade broda SAUCY SUE

SHIP	=	SAUCY SUE
RANK	=	CAPTAIN
SNAME	=	BOWLINE
RANK	=	FIRST MATE
SNAME	=	SCUPPER
RANK	=	SECOND MATE
SNAME	=	HOOK

U registru posade nađeno je prezime BOWLINE i ime broda. BROD – SAUCY SUE traži se luka gdje je pristao – RIO DE JANEIRO

SHIP = SAUCY SUE
PORT = RIO DE JANEIRO

Sada se kapetanu Bowlineu može dostaviti sretna vijest. Ovdje se radi o vrlo skromnoj bazi podataka, jer svako pitanje vodi samo do jednog odgovora. U ozbiljnijem poslu postoji mnogo složenija baza podataka i svako pitanje vodi do više odgovora.

adresu. Osim toga, možda ne želite da u svakom zapisu budu pohranjeni svi podaci o robi koju ste mu prodali. Zamislite da ste mu 15. siječnja prodali tucet cipela „model 46", a ima zaista mnogo podataka o tom modelu: napravljen je od bukovine, gornji dio je od kože s mjedenim čavlima na obodu i svaki par teži 7 kg. Vama se ne da pri svakoj prodaji iznova sve pisati. Zato svaki poslovni zapis mora biti povezan sa zapisom u kojem su svi podaci o cipelama tog modela, koliko ih ima na skladištu i koliko ih je naručeno, a ne treba ni reći da može biti mnogo ovakvih zapisa o proizvođačima cipela s kojima surađujete.

Sve to može izgledati jednostavno, ali su mnogi stručnjaci o tome napisali knjige i stekli ugled istražujući taj problem. U programskom paketu superfeji o kojem je ovdje riječ, dva zapisa povezuju se tako da se u njih uključe isti stavci. Na primjar, ako želite povezati zapis s osobnim podacima stranke i onaj o njezinim transakcijama, treba ispod etikete samo dopisati broj za identifikaciju, recimo ID = 34, pa će taj stavak ući u svaki od poslovnih zapisa. Tako se mogu naći poslovne transakcije gosp. Smitha kad se nađe njegov osobni zapis i u njemu ID (broj identifikacija) po kojem se onda nalazi sve ostalo.

Postupak je poznat u trgovačkom svijetu kao povezana baza podataka (Relational database), koju se na primjer može upitati: „U kojem su gradu prodavane cipele što su za nas izrađene u tvornici X?" To praktički znači da treba ići po stupnjevima: najprije pogledati u zapis o tvornicama, naći tvornicu X, ustanoviti kakve cipele proizvodi, pogledati u zapise o transakcijama i ustanoviti koji su kupci nabavili cipele proizvedene u toj tvornici, i konačno u adresne zapise da se ustanove gradovi gdje kupci žive.

Sve to može biti dosta komplicirano, iako u slučaju tvornice cipela, po svoj prilici, nema u pojedinim stupnjevima tako mnogo odgovora. Ali zamislite da ste lutajući reporter koji istražuje malverzacije u nekoliko kompanija sa zajedničkom bazom podataka. Istraživanje počinjate s kompanijom „Prijevi Poslovi & co." i pitate: „koje kompanije imaju iste osobe na popisu direktora?", zatim „da li se iste osobe nalaze na popisu direktora još nekih kompanija?"

Sumnja se da postoji direktorski lanac koji seže sve do neke tajanstvene firme u Panami. Ali da biste razmrsili klupko, treba vrlo pažljivo pretražiti podatke tisuća kompanija i tada se otkrivaju još čudnije stvari. Kompanija „Prijevi poslovi" ima pet direktora: A, B, C, D i E. Direktor A je istovremeno direktor u još tri druge kompanije – „Magla", „Prekobrojna" i „Okus šinne", od kojih svaka ima tri do deset direktora koji su ujedno direktori i nekih drugih kompanija. Provjeravajući direktore A, sumnjive je postala kompanija „Magla" s direktorima F, G, H i I. Nastavljate istraživanje s direktorom F, koji je također direktor u još pet kompanija od kojih svaka može biti

začetnik malverzacije. Treba svaku od njih provjeriti, ustanoviti u kakvoj su vezi s Prljavim poslovima i ... Tim smo primjerom prikazali samo dvoslojnu istragu, dok se cilj slične pretrage može nalaziti i desetak slojeva dalje.

Primjer pokazuje da često spominjena tvrdnja „sve se nalazi u kompjutoru" može imati i vrlo realnu podlogu i da se sve ovo može vrlo lako pretvoriti u kompjutorsku eksploziju, posljedica koje je stvaranje više informacija nego što ih strojevi i njihovi korisnici mogu obraditi (o čemu će biti još govora na str. 175). Već se sada događa da baza podataka akumulira više podataka nego što je potrebno.

Za razliku od papirnih akata na kojima zapisane informacije miruju u svojoj krutosti i nedohvatljivosti, kompjutorske informacije nezadrživo zuje uokolo, umnažaju se i često iz jednostavnih poslovnih postupaka stvaraju teške intelektualne probleme.

Obrada teksta

Druga standardna vrst papirnih dokumenata je ona koja uglavnom sadrži tekst: to su pisma, članci, izvještaji, memorandumi i knjige. Programski postupak koji na kompjutorima radi taj posao naziva se 'obrada teksta' (word processor). Pisanje teksta postoji već mnogo stotina godina i spada u dobro proučena područja ljudske djelatnosti, pa zato nema mnogo prostora za originalnost i novotarije u kompjutorskoj verziji tog prastarog postupka.

Svi programski paketi za obradu teksta omogućuju korisniku da napiše i izvrši sve ispravke na ekranu, da ga otisne pomoću printera na papir i zatim ga pohrani na disk, spremnog za dalju upotrebu.

Glavna prednost postupka sadržana je u tome što se tekst (poput ove knjige) ispisuje na ekranu, ispravlja i prepravlja, otiskuje na printeru, opet ispravlja, a da se svaki put iznova ne mora obavljati mučno pretipkavanje. Standardna pisma, izvještaji i ugovori (ili samo njihovi dijelovi) mogu se spremati na disk ili vaditi s njega slobodno mijenjajući pojedine odjeljke sve do njihova konačna oblika.

Pogledajmo redom sve tri faze. Najprije se pomoću tastature kakvu ima svaki pisaći stroj piše tekst na ekranu. Za to vrijeme mogu se mijenjati ili brisati pojedini dijelovi teksta, premještati i zamjenjivati odlomci, pronalaziti pogrešno napisane riječi ('kompjutter') i ispravljati ih ('kompjutor'). Mogu se sa diska ubacivati neke standardne formulacije ili natpisi i dodavati ih već napisanom tekstu. Treba samo upitati nekog daktilografa koliko se tu uštedi vremena.

Posebno pametni softverski paketi dozvoljavaju istovremeno manipuliranje s nekoliko različitih tekstova, no ne baš sasvim doslovno, ali u kompjutorovoj memoriji je naporedo više tekstova koji se mogu naizmjence obrađivati. Neki veći kompjutorski sustavi imaju ekran podijeljen na dva ili više dijelova i na svakom se dijelu može posebno obrađivati određeni dio teksta.

Kad se softverski paketi za obradu teksta vrte na standardnim kompjutorima, korisnik se mora služiti standardnom tastaturom i tipkama. To nije uvijek spretno, jer se često naredbe za pomicanje kursora po ekranu daju pritiskom na CTR (control) tipku i jedno slovo, a ima i komandi za koje treba pritisnuti i tri tipke za redom. Zato su kompjutori specijalno konstruirani za obradu teksta opremljeni posebnim vrstama ekrana i tipaka na tastaturi.

Bilo bi lijepo kad bi se na ekranu mogao složiti tekst jednako kao da je otisnut na papiru. Na žalost, to je rijetko kad izvedivo, jer bolji printeri otiskuju slova proporcionalnom metodom, to jest svakom slovu daju onoliko prostora koliko mu zaista treba — slovo l ili j zauzima proporcionalno uži prostor od, recimo, slova W ili m — dok na ekranu sva slova zauzimaju jednaki prostor, bez obzira na širinu. Nadalje, malo je ekrana koji u jednom retku ispisuju više od 80 znakova (na jeftinijim izvedbama samo 40), dok većina printera otiskuje 132 ili više znakova u retku. Zbog toga onaj propagandni slogan kojim se služe mnogi proizvođači, „ono što vidite na ekranu, to ćete dobiti i na papiru" — na žalost nije istinit.

Druga faza je spremanje napisanog teksta na disk, što bi najbolje bilo da se provodi automatski, tako da korisnik nema s tim nikakve brige. U dobro napisanim programima za obradu teksta, pomicanjem kursora tekst se jednostavno premješta s ekrana na disk ili obratno, pa se može obrađivati tekst i od nekoliko milijuna znakova. Program bi osim toga morao sukcesivno i automatski bilježiti na disk sve što se piše, te u slučaju da nestane napajanja strujom, cijeli posao, koji je možda trajao i nekoliko sati, ne može biti jednostavno izgubljen.

Treća faza je tiskanje napisanog teksta. Kao što smo već ustanovili, ono što vidimo na ekranu ne može se u potpunosti prenijeti na papir. Na primjer, pišući ovu knjigu, nisam znao kako će izgledati otisnuti tekst, koliko će znakova slagar složiti u jednom retku ili koliko će praznih redaka ostaviti između odlomaka. Moj softver za obradu teksta to obavlja umetanjem kontrolnih znakova u sâm tekst, tako da na početku reda stoji točka iza koje slijede dva velika slova i dva broja, na primjer .LL49. Printer će to shvatiti kao red od 49 znakova (poput retka u ovoj knjizi) i čim dođe do početka neke riječi koja dužinom prelazi ovu mjeru, automatski će je premjestiti u slijedeći red. Kad treba izravnati i desnu marginu (kao u ovoj knjizi), tekst treba staviti u memoriju i dodati razmake između riječi, tako da svaki redak bude dugačak točno 49 znakova.

Nema razloga da ovakvi programi ne budu još pametniji, te da rješavaju i neke druge zadatke osim formatiranja teksta ili preslikavanja na papir točno onako kako je prikazan na

ekranu (Vidi 'Lisa' str 46-47) Neki programski paketi mogu automatski numerirati stranice ili odjeljke ispisivati tabulirane liste i izvoditi sve one kozmetičke manipulacije na tekstu koje bi radila jedna dobra tajnica.

Da se ne zavaravamo pisanje programa nije uopće jednostavan posao pa se zato redovito kupuju kod profesionalnih proizvođača softvera

Knjigovodstvo

Dvojno knjigovodstvo počelo se voditi već u 13 stoljeću u Italiji a princip se nije promijenio sve do danas pa nije drugačiji ni u kompjutorskom sustavu Danas, naime postoji softver koji simulira knjigovodstvenu karticu s iscrtanim redovima i stupcima upisanim svotama u kućice (npr za mjesečnu prodaju) izračunava porez trgovačku maržu i ukupne troškove, da bi na kraju iskazao dobit Sve je to korisno za rutinsko obračunavanje proteklog poslovanja a i za procjenu budućih uvjeta Što će se dogoditi ako porezi narastu za 3 % troškovi za energiju za 15 % (ili 25 %) a osobni dohoci za 4 9 %? Umjesto da se uz pomoć džepnog kalkulatora mučno probijamo kroz stotine matematičkih operacija dovoljno je da u program ubacimo jednu sumu a sve ostalo prepustimo kompjutoru

I opet moram ponoviti ono što sam već nekoliko puta rekao softverski paketi omogućuju da se bavimo kreativnijim poslovima

Knjigovodstvene kartice nije uvijek najbolji način predočavanja numeričkih podataka posebno onda kad se traže odnosi između raznih skupova veličina Neki proizvođač na primjer želi saznati utjecaj reklamne akcije na prodaju svog proizvoda Kompjutor će iskazati sve brojčane podatke ali na teško razumljiv način pa se stoga sve više upotrebljavaju standardni softverski paketi koji rezultate neke akcije iskazuju umjesto brojevima grafičkim prikazima Operacija je jednostavna označe se samo dvije varijable — troškovi recimo na horizontalnoj a vrijeme na vertikalnoj osi — i program će samostalno pronaći najmanje i najveće vrijednosti prenijeti ih u odgovarajućem mjerilu na grafikon te prikazati na ekranu i otisnuti na printeru Ne treba niti spomenuti da je moguće istodobno prikazivanje i više od dviju vrijednosti

Postoji još jedan način prikazivanja poslovnih informacija a to je prikaz u obliku torte (pie chart) koji rezultate poslovanja prikazuje u kružnom obliku Što je grafički hardver bolji i savršeniji sa što više piksela i boja (vidi str 30-31) to grafičke metode prikazivanja poslovanja postaju sve svakodnevnije

Računovodstvo

Kompjutor ne griješi (ili barem ne bi trebalo da griješi) u aritmetici i postupcima ne zaboravlja

Kompjutor i njegov ekran mogu zamijeniti pisaći stroj i papir Za to služi softverski paket za obradu teksta (Word Processor) Njegova je velika prednost u tome što zbog eventualnih ispravaka ili prestilizacije ne treba iznova tipkati cijeli tekst
1 Promijenjena je duljina retka sa 45 znakova u retku (A) na 32 (B)
2 Ukopčana je naredba PAKOVANJE koja prebrojava znakove u svakom retku te sve što prelazi mjeru (32 znaka u jednom retku) prebacuje u novi red
3 Kursor se nalazi na prvoj pogreški (kompuler)
4 Korisnik će u pogrešno napisanu riječ ubaciti slog ju izbrisati slovo e i na isto mjesto upisati o — kompjutor
5 Ako se pogreška ponavlja kao ovdje onda je još jednostavnije dati naredbu ZAMIJENI RIJEČ koja će ispraviti pogrešku ma koliko puta se ponavljala (6)
7 Posljednji pasus u primjeru (A) prebačen je na drugo mjesto u primjeru (B) Kada je tekst gotov može se pohraniti na disk ili vrpcu i pozvati ga svaki put kad bude trebalo otisnuti novu kopiju ili štogod promijeniti Softver može biti predviđen tako da vadi samo pojedine fajlove sa diska i da pomoću standardnih formulacija načini više raznih dokumenata Drugi dio programa Izv formator kontrolira tiskanje teksta na printeru Njime se bira običan ili proporcionalni slog izravnava lijeva i desna margina numeriraju pasusi i stranice sastavlja indekse i sadržaje knjiga

poslove koji su mu naređeni. Osim toga on mnogo lakše barata brojevima nego većina ljudi, pa je posve razumljivo da ga se uposli na računovodstvenim poslovima, za što postoje već mnogi softverski paketi.

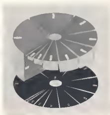
Promet robe je u osnovi vrlo jednostavna stvar. Netko prodaje kikiriki, a netko drugi softverske pakete, ali u jedno i u drugo treba uložiti novac koji će se nakon prodaje vratiti. Razlika između uložena i zarađena novca je ono što će ulagaču omogućiti da provede odmor na Bahamskim otocima, pa on zbog toga odmah želi znati hoće li otići na odmor ili ne, a to će mu reći njegov kompjutor koji stalno prati iznose troškova i prihoda te jednostavnim oduzimanjem prikazuje trenutno stanje.

Knjigovodstvena bilanca talijanske tvrtke Francesco Datini koja je poslovala u Barceloni 1399.

Dakako, ovo je ipak pojednostavljena slika knjigovodstva i dovoljno je samo baciti pogled na stručnu literaturu koja se bavi ovim problemom, da se vidi kako se knjigovodstvo stoljećima razvijalo da bi danas doseglo zavidan stupanj kompliciranosti.

Najjednostavnije je predodžbu o poslovanju steći analizom serije ugovora. Svaki ugovor prolazi kroz brojne etape i svaka se mora posebno registrirati. Treba izračunati porez na promet i porez na dohodak, što se izračunava u različitim fazama poslovanja. Treba provjeriti sve poslove s gosp. Smithom, sve što mu je prodano u četvrtom kvartalu i sve ono što je utrošeno na osobne dohotke i ekonomsku propagandu.

U osnovi je kompjutoru to vrlo jednostavno ako su mu podaci podastrti na odgovarajući kompjutorski način. Važno je da svaki ugovor sa svim informacijama bude registriran u bazi podataka, odakle se može izvući sve što je potrebno. Na žalost, u računovodstvima se obično tako ne radi, a vidjet će se i zašto. Oslanjanje na tehnologiju tinte i papira nameće dva ograničenja. Prvo, vidljiv je samo jedan dio informacije — ime dobavljača ako su registri složeni po abecedi, a datum nabavke ako su poslagani kronološki — dakle vidljivo je jedno ili drugo, ali nikada oboje zajedno. Stoga u papirnim sustavima mora postojati i dnevnik poslovanja i salda-konto. Drugo, u papirnim sustavima moguće su pogreške tokom čitavog postupka, što je glavni razlog zbog kojega se vodi dvostruko knjigovodstvo, dok u kompjutorskom sustavu to nije potrebno. Dnevnik poslovanja i salda-konto su samo dva različita načina uporabe istih podataka, a ta se razlika očituje u postupku pretraživanja baze podataka: hoće li se pretraživati prema imenu ili prema datumu. Sve dok se ne pojavi pogreška u unošenju inicijalnih podataka, nije potrebno dvostruko knjigovodstvo, jer kompjutor, u to treba vjerovati, neće pogriješiti.



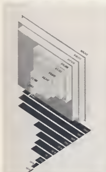


Za kompjutore s ekranima u boji ima softvera koji u grafici visoke rezolucije mogu prikazati poslovne trendove. U poslovnom svijetu u modi su grafikoni (u obliku stupića i torti). Takav način prikazivanja pregledniji je za one koje ne zanimaju detalji već samo opća slika.

Za takvo grafičko prikazivanje u engleskom su udomaćeni termini bar (prečka) i pie (pita). Prim. prev.

Šteta je što su programerima koji su sastavljali prve računovodstvene programe smetali knjigovođe zahtijevajući u skladu sa svojim pa

pirnim navikama pristup u svaku fazu programa čak i na onim mjestima gdje je to gotovo nemoguće. To je razlog zbog kojega kompjutorski knjigovodstveni sustavi teže da čak i u nekim sasvim besmislenim situacijama zadrže stil papirnog knjigovodstva koji što potpunije odgovara vanjskom izgledu poslovnih knjiga i salda-konta. Korisnici su zbog toga prisiljeni da dva puta unose iste podatke, pa se razlog što je ovaj softver tako skup i tako neelastičan nalazi upravo u toj činjenici.



Na ovom kompjutorskom iskazu prikazan je financijski plan vrlo male poslovne organizacije, kako bi se ustanovilo koliko robnih jedinica treba prodati u prvom kvartalu. Time se ujedno određuje broj zaposlenih, troškovi proizvodnje i distribucije i naravno zarada. Zadnji red pokazuje što se događa s bankovnim računom vlasnika koji na početku poslovanja počinje s dugom od 20 000 $. Takav kompjutorski iskaz omogućuje vlasniku eksperimentiranje u planiranju: ako krene s prodajom od samo 4 500 jedinica po kvartalu (A), deficit je siguran, ako pak može isporučiti 8 000, može se nadati uspjehu.

Na slici C vidi se kako je organiziran kompjutorski iskaz softvera Microsoft Multiplan.

1 Rubrike se nalaze po moću brojeva koji označuju horizontalne redove i vertikalne stupce.
2 Iznos robnih jedinica. Sve ostalo izračunava se iz ove sume.
3 Predviđa se da će prodaja rasti po stopi od 10% svakog idućeg kvartala. Formula [RC—1]*1.1 znači: unesi iznos u isti red lijevoga stupca i pomnoži ga s 1.1. Ta je formula kopirana u svaku rubriku s desne strane, a sviha joj je ponoviti istu operaciju u svakom kvartalu.
4 Broj zaposlenih iznosi 1 + iznos iz reda 14 istoga stupca (robne jedinice) podijeljen sa 1 000.
5 Osobni dohoci i režijski troškovi kalkulirani su prema broju zaposlenih i 7 000 odnosno 4 000 za svaki kvartal.
6 Troškovi repromaterijala i opreme kalkulirani prema broju robnih jedinica.
7 Ukupni troškovi — zbroj iznosa u redovima 7 do 10 u istom stupcu.
8 Prihod od 19 $ po robnoj jedinici.
9 Kvartalna dobit — prihod umanjen za visinu troškova.
10 Bankovni saldo počinje s −20 000. Svaki sljedeći saldo jednak je saldu prošlog kvartala povećanom za iznos prometa, također iz prošloga kvartala.

A

Joe Soap Inc.

Quarter:	1	2	3	4
EXPENSES				
Employees	6	6	6	7
Wages	30000	41060	46119	40027
Overheads	22000	23000	25700	27000
Materials	22000	24750	27225	30046
P & P	4000	4000	5445	6000
Total Paid	87500	95100	100606	113022
Units Sold	4000	4000	5441	5000
INCOME				
Sales	76000	84050	100456	113801
Quarterly mvt.	−2000	−1100	[illegible]	820
Bank Balance	−20000	−22000	−23100	−22230

B

Joe Soap Inc.

Quarter:	1	2	3	4
EXPENSES				
Employees	9	10	11	12
Wages	63000	70000	74700	81536
Overheads	36000	36000	42738	46502
Materials	44000	44020	48400	53240
P & P	8000	8800	9680	10648
Total Paid	147000	160000	175000	180010
Units Sold	8000	8800	9680	10648
INCOME				
Sales	152000	167200	[illegible]	202312
Quarterly mvt.	5000	9000	8300	10000
Bank Balance	−20000	−15000	−6400	−40

C

	1	2	3	4	5
1	"Joe Soap Inc"				
2					
3	"Quarters"	"1"	"2"	"3"	"4"
4	"EXPENSES"				
5	"Employees"	1+(R14 C)/1000	1+(R14 C)/1000	1+(R14 C)/1000	1+(R14 C)/1000
6	"----------"	"----------"	"----------"	"----------"	"----------"
7	"Wages"	(R5 C)*7000	(R5 C)*7000	(R5 C)*7000	(R5 C)*7000
8	"Overheads"	(R5 C)*4000	(R5 C)*4000	(R5 C)*4000	(R5 C)*4000
9	"Materials"	(R14 C)*5	(R14 C)*5	(R14 C)*5	(R14 C)*5
10	"P&P"	(R14 C)*1	(R14 C)*1	(R14 C)*1	(R14 C)*1
11	"----------"	"----------"	"----------"	"----------"	"----------"
12	"Total Paid"	SUM(R7 C:R10 C)	SUM(R7 C:R10 C)	SUM(R7 C:R10 C)	SUM(R7 C:R10 C)
13	"=========="	"=========="	"=========="	"=========="	"=========="
14	"Units Sold"	8000	(RC[-1])*1.1	(RC[-1])*1.1	(RC[-1])*1.1
15	"----------"	"----------"	"----------"	"----------"	"----------"
16	"INCOME"				
17	"Sales"	(R14 C)*19	(R14 C)*19	(R14 C)*19	(R14 C)*19
18	"=========="	"=========="	"=========="	"=========="	"=========="
19	"Quarterly mvt"	R17 C-R12 C	R17 C-R12 C	R17 C-R12 C	R17 C-R12 C
20	"Bank Balance"	−20000	R[-1]C[-1]+RC[-1]	R[-1]C[-1]+RC[-1]	R[-1]C[-1]+RC[-1]

CAD kompjutor (Computer Aided Design — kompjutorsko generiranje slike) Operator ima ispred sebe jedan konvencionalni ekran (lijevo) za programske liste i drugi u boji za grafiku visoke rezolucije (desno). Na stolu je tastatura i ploče za digitalizaciju

Na suprotnoj strani
Gore kompjutorskom grafikom mogu se predstaviti neke informacije koje bi inače bile teško shvatljive. Na ovim četirima slikama prikazan je svjetlosni intenzitet dvostruke galaksije M51. Slike su snimljene vrlo dugačkom ekspozicijom. CAD je svaku razložio na 4 milijuna piksela i transformirao u slike koje su sličnije nekom planinskom pejzažu. Svaki piksel ima svoju šifru boje, a njihove relativne visine predstavlje svjetloću svake pojedine zvijezde u galaksiji što nalikuje oštrim šiljcima planinskih vrhunaca

Dolje Iz razrađenog tehničkog nacrta CAD može generirati trodimenzionalnu sliku mehaničkog sklopa u sastavljenom ili rastavljenom stanju. Pored toga može dati upute o alatu za izradu dijelova. Te se informacije kasnije mogu upotrijebiti za izradu tehničke dokumentacije, kataloga i tehničkih uputa

Dolje desno sa CAD emulacijom ne prikazuju se samo dijelovi nacrta, već se također mogu izračunati i opterećenja pri radu, pokazati odstupanja, odrediti zazore i izračunati potrebna količina materijala. Inače se ni jedna od tih informacija ne bi mogle dobiti prije nego što se izradi prototip

Primjena kompjutora doživjela je fascinantan razvoj na području slika i crteža. Ranije su kompjutori korišteni kao pomoć tehničkim crtačima, zatim se prešlo na generiranja nizova slika za simulatore u obuci pilota, pa kao pomoć u proizvodnji crtanog i u montaži snimanog filma. Uz to su kompjutori upotrebljavani za stvaranja posebnih efekata na televiziji i filmu, za analizu satelitskih snimaka Zemljine površine i podataka o dalekim galaksijama, za kreiranje uzoraka tekstila te za izvedbu perspektivnih crteža u arhitekturi. Liječnici se također služe kompjutorima za analiziranje rentgenskih snimaka, skenerskih snimaka mozga i za različite ergonometričke i dinamička analize ustrojstva ljudskoga tijela

Druga linija razvoja tekla je prema hardvaru i softveru koji će 'gledati' pomoću televizijske kamere. Svrha je bila stvoriti napravu koja će razaznavati dijelove neke veće cjeline

Prije su sva ta područja djelovala samostalno i odvojeno, ali danas se granice sve više gube. Na slijedećih sedamnaest stranica pokušat ćemo prikazati koliko je široko područje kompjutorskog dizajniranja i opće likovne aplikacije

Na tom polju amater nije u sretnu položaju jer većina likovnih aplikacija zahtijeva u prvom redu moćan hardver. Za dobru vizualizaciju potrebna je grafika visoke rezolucije, a to znači mnogo RAMa, ekrani visoke rezolucije i moćni procesori. (Vjerojatno će slijedeća generacija vizualnih kompjutora koristiti tehniku paralelne obrade podataka, opisanu na str. 174-175.) Prvo računalo koje se tome približava je Appleova Lisa, ali ni ona nije dovoljno moćna, a niti dovoljno jeftina, da bi pravu likovnu aplikaciju učinila dostupnom širokom krugu korisnika. Prikladna rješenja pojavit će se, bez sumnje, tokom idućih pet godina

Kompjutorska slika može se dobiti na dva načina. Prvi kao osnovu koristi tehnički crtež, skicu ili fotografiju, a kompjutor je samo pomoć crtaču ili umjetniku. Drugi način je da kompjutor prisilimo gledati kao što mi gledamo i da dokaže kako zaista 'vidi', kompjutor može izraditi sliku onoga što je vidio

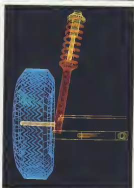
Unutar ove podjele moramo naznačiti još jednu osobinu: ono što kompjutor prikazuje na ekranu nije ni izdaleka sve što može učiniti. Na primjer, u konstrukcijskom odjelu neke tvornice motora CAD kompjutor (Computer-Aided Design — crtanje pomoću kompjutora) koristi se pri projektiranju mjenjačke kutije. Na ekranu su prikazani svi detalji zubaca na zupčanicima i ako je potrebno analizira se opterećanje pojedinih dijelova sklopa, ustanovljava njihove čvrstoća i proračunava količina materijala za njihovu izradu uz optimalnu štednju materijala i vremena izrade

Kompjutor u potpunosti obavlja ulogu konstruktorova pomoćnika i poslušno izvršava naredbe poput 'nacrtaj ovdje kružnicu, tamo povuci crtu, izračunaj naprezanja ovoga dijela' i prikazuje na ekranu model konstrukcije koji se inače izrađuje od drva, kako bi se vidjelo da li sve dimenzije odgovaraju. Kad je to obavio, kompjutor će poslati upute kompjutorski upravljanim strojevima da izrade pojedine dijelove: istokari ovo na ovakav promjer, izbrusi ovaj dio da bude ravan, probuši ovdje rupu od 9mm

Kompjutor će također iskalkulirati veličine i oblike metalnih otkivaka potrebnih za izradu svakog pojedinog dijela, a u savršenijim izvedbama sve će pojedinosti o količinama materijala proslijediti do kompjutora koji vodi knjigovodstvo i narudžba materijala







Slike na ovoj strani
Jednostavnim pritiskom na tipku kompjutor može mijenjati boje na ekranu. Ta je osobina vrlo upotrebljiva pri dizajniranju sagova.

Engleski proizvođač cipela Clark i odio CAD u Cambridgeu usavršili su tehniku generiranja slika koje se jedva razlikuju od gotova proizvoda. Taj postupak ne štedi samo vrijeme već smanjuje i velike troškove za izradu nesigurnih modnih prototipova. Treba primijetiti kako je sasvim nestao onaj karakteristični poput fasete izlomljeni oblik kružnih linija koji je bio neizbježan u starijim kompjutorom generiranim trodimenzionalnim grafikama. Ta se slabost jasno vidi na slici aviona Harrier na sl. 105.

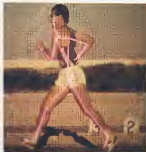
Nacrt kemijskog postrojenja može se brzo pretvoriti u perspektivnu skicu. Što crtačima uštеđuje na stotine sati strpljiva rada. Kompjutorom generirane slike zgrade iz arhitektonskog nacrta gotovo se ne razlikuje od snimke načinjene nakon završene gradnje.

Slike na suprotnoj strani
Strukturalna slika aviona Tornado skoro je isto tako lijepa kao i sam avion. Na njoj se vidi 5 glavnih cjelina letjelice: radar (ružičasto), trup (plavo), krila (narančasto), motori (zeleno) i rep (crveno).

Važno područje dizajna su ergonomske studije gdje kompjutorom generirani crteži zamjenjuju drvene modele.

Znanstvenici proučavaju kretanje sportaša tako da u filmski snimak trkača rukom ucrtaju shemu udova, a računalo prema tome izračunava dinamiku pokreta.





Najčešću primjenu kompjutorske grafike nalazimo u području simulacije. Civilne zrakoplovne kompanije, mornarica i vojno zrakoplovstvo smatraju da je preskupo obučavati posade na pravim avionima, helikopterima, brodovima i tenkovima. Simulator za pilote lovca, na primjer, u potpunosti imitira ponašanje njegova aviona. Kompjutor reproducira sve karakteristike upravljanja na različitim visinama i pri raznim brzinama, s punim ili praznim rezervoarom i spremištem municije. Simulacija leta na avionu *Harrier* (proizvod Redifussion) tako je savršena da se avion ponaša kao pravi. Na slici ne manjkaju ni takve pojedinosti kao što su zasjenjenja ispod trupa i pod krilima i pogled na Zemlju što se gubi u dalekoj izmaglici. Zahtjev za prirodnom brzinom kretanja (real time) natjerao je programere da se zadovolje s isjenčanim oblinama (to se vidi na uglastoj liniji kokpita), što se moglo izbjeći kad se slike ne bi morale izmjenjivati s frekvencijom od 25 slika na sekundu.

Dolje: Tri pogleda na sletno-poletnu stazu. Dva sa istoga mjesta, od kojih je prvi za lijepa, a drugi za maglovita vremena. Na trećoj slici je ista zračna luka, ali noću i sa udaljenosti od nekoliko kilometara. Takvi kompjutorski programi postoje za većinu značajnijih svjetskih zrakoplovnih luka.

Desno gore: Mnogo je brže i jeftinije ispitati projekt nove saobraćajnice pomoću kompjutora, nego onda kada je već sagrađena ustanoviti da ne odgovara.

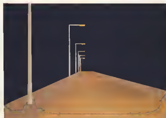
U sredini i dolje desno: Kompjutorske simulacije koriste se i za rješavanje sasvim neočekivanih problema. Tako, na primjer, laboratorij za cestovna istraživanja britanske vlade pomoću kompjutorske simulacije projektira cestovnu rasvjetu, što je dakako mnogo jeftinije nego ustanoviti greške na već izvedenom projektu. Promjena u nosačima rasvjetnih tijela, što je samo mala promjena u kompjutorskom programu, daje mnogo ravnomjerniju raspodjelu svjetla na kolniku.











OBRADA SLIKE

Kompjutor može biti vrlo konstan ako se upotrebljava za poboljšavanje i manipulaciju fotografske slike što je na pola puta do istinskog gledanja kompjutorom. On naime može od neoštre mutne slike načiniti oštru prirodne boje pretvoriti u natprirodne ili na vrlo preciznim satelitskim snimcima označiti ležišta rude nafte ili položaje projektila. Veći dio rješavanja ovih zadataka osniva se na otkriću koje je načinio još početkom 19. st. francuski matematičar i fizičar Jean Baptiste Joseph Fourier (1768-1830).

Prije nego li podrobnije opišemo Fourierovo otkriće treba objasniti na koji način kompjutor uopće može 'vidjeti'. Da bi se to postiglo sliku treba najprije digitalizirati tj. pretvoriti je u bitove u jedinice i nule. Najpogodnija za to je dakako, televizijske slika koju treba podijeliti u piksele (vidi str. 30-31), izmjeriti svjetlosni intenzitet svakog piksela pa ako se radi o slici u boji treba još izmjeriti i svjetlosnu razinu svake od triju osnovnih boja. Ali da bi stvari bile jednostavnije za sada pretpostavimo da se radi o crnobijeloj slici koju televizijska kamera ionako rastavlja na točke i pretvara točke u napon te sve što još treba učiniti je da se ti naponi pretvore u brojeve pomoću analogno-digitalnog konvertora ili pretvarača (ADC).

Televizijska kamera očitava sliku liniju po liniju skaniranjem. Pretpostavimo da želimo imati toliko redova piksela koliko ima linija televizijska slika (oko 200 u industrijskom tipu kamere). Ako ADC radi takvom brzinom da uspije nepreviti 200 pretvorbi po liniji dobit ćemo sliku od 200 × 200 piksela.

Sljedeće je pitanje s koliko svjetlosnih razina želimo raditi. ADC pretvarači mjere intenzitet svjetla i pretvaraju ga u bitove a od broja bitova ovisi finoća mjerenja. Najgrublje mjerenje je s jednim bitom (kad nula znači mrak, a jedinica svjetlo) dok veću točnost, 16 svjetlosnih razina daje 4-bitna pretvorbe a 8-bitna 256 što je već vrlo velika točnost. Koliki ćemo stupanj točnosti odabrati ovisi o brzini i kapacitetu memorije računala.

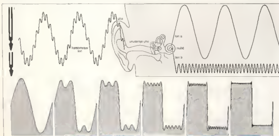
Čin pohranjivanja slike u memoriju računala predstavlja zapravo svođenje dvodimenzionalnog predloška na jednu dimenziju.

Da bismo lakše shvatili Fourierovu oštroumnu analizu poslužimo se još jednom jednodimenzionalnom pojavom — glazbom. Čisti glazbeni zvuk sastoji se od sinusnoga vala određene frekvencije a najbolji primjer je zvučna viljuška. Kad istovremeno zvuče dvije zvučne viljuške čuju se istovremeno dva tona iako zračni tlak na uho ne može istom trenutku imati dva različita iznose. Uho dakle mora pomoću Fourierove analize izdvojiti dva sinusna vala iz jednog vala složenog oblika.

Prema tome jasno je da se zvuk orkestra u kojem istodobno zvuči nekoliko desetaka tonova različita valna oblika i frekvencije može razlučiti u čiste sinusne tonove. Isto se može učiniti i se bilo kojim zvukom ili valnim oblikom. Fourier uči da se svake funkcija F (x) može pretvoriti u sinusne i kosinusne valove prema jednadžbi

$$F(x) = a_0 + a_1\cos(x) + b_1\sin(x) + \ldots a_n\cos(nx) + b_n\sin(nx)$$

gdje su a_n i b_n Fourierovi koeficijenti. Posebno je teško predočiti kako jedan pravokutni valni oblik (poput onoga na slici dolje koji je u stvari niz binarnih jedinica) može biti sastavljen zbrajanjem određenog broja kontinuiranih sinusnih valove. Prema Fourierovoj analizi sastavljen je od osnovnog vala, čije je poluperioda jednaka širini impulsa plus val od $\frac{1}{3}$ amplitude i dužine, plus val od $\frac{1}{5}$ amplitude i dužine ... itd.



Na slici vidimo kako se dva sinusoidalna zvuka različite frekvencije sastavljaju u jedinstven zvučni val. U pužiću unutarnjeg uha zvukovi se razdvajaju a matematičari kažu da to uho provodi Fourierovu analizu zvuka.

Dolje. Najčudnije izgleda da je jedan pravokutni impuls zapravo sastavljen od mnogo sinusnih impulsa: čisti sinus plus $\frac{1}{3}$ sinusa uz 3 puta veću frekvenciju plus $\frac{1}{5}$ sinusa i 5 puta veću frekvenciju plus ... konačno nastaje čisti pravokutni val. Kompjutor može postupak izvesti u obrnutom smjeru što na primjer znači da se digitalizirana fotografija može rastaviti na sinusne valove s kojima se mogu vršiti razne manipulacije. Time je otvoren put kompjutoru da može na mnogo načina intervenirati u slici.

Gore lijevo: Zračni i satelitski snimci mogu se digitalizirati i kompjutorski obraditi pa se na takvim snimcima razaznaje mnogo više detalja nego na originalnim. Lijeva gornja slika snimljena je na običnom, a desna na infracrvenom fotografskom materijalu. Dolje desno kompjutor je analizirao i klasificirao pojedina gradska područja. Dolje lijevo kompjutorska slika statističke analize različitih gradskih područja.

Lijevo: simulacija unutrašnjosti ljudske glave. David Hogg sa Sveučilišta Sussex koristi snimke cerebralnog skanera i pomoću kompjutorskog programa koji zna sve o strukturi ljudske glave traga za eventualnim kanceroznim pojavama.

Gore: snimak dvaju pršljenova oboljele kralješnice sa živcima i diskusom.

Sasvim gore: snimak površine Marsa nakon što je prošao kroz više kompjutora. Izvorna TV slika odaslana je preko radija, pohranjena u kompjutorsku memoriju, ispravljena i poboljšana, pa tek onda fotografski snimljena sa kompjutorskog ekrana.

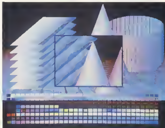
Fourierove matematičke transformacije nisu ograničene samo na jednodimenzionalne pojave poput zvuka, već se mogu primijeniti i na dvodimenzionalne pojave, poput slika.

Brojni postupci osnovani na Fourierovim transformacijama omogućuju poboljšavanje kvalitete digitaliziranih fotografija. Povećanje oštrine postiže se izdizanjem visokih frekvencija (poput izdizanja visokih tonova u audiotehnici), od kojih su formirani rubni dijelovi objekata na slici. Ako se frekvencije pojačaju, rubovi predmeta će biti vidljiviji, a slika oštrija i briljantnija.

SLIKANJE S BROJEVIMA

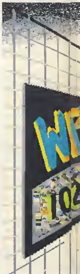
Slikar istiskuje boju iz tube i kistom je nanosi na platno. Kompjutor uzima boje sa tablice boja i pomoću programa nanosi ih na ekran. Nema razloga da kompjutor ne postane neka vrst inteligentnog slikarskog platna.

Većina slika sastoji se od kontura unutar kojih se nanosi boja. Rekli bismo: to je isto što i slikanje pomoću kompjutora. Uostalom, tako je nastala i većina tradicionalnih umjetničkih djela. Pri slikanju s kompjutorom najteži je prvi korak: unošenje osnovnog crteža u memoriju. Umjetnik doduše ima na raspolaganju tastaturu kompjutora i pomoći tipaka može voditi svoju 'olovku' po ekranu, ali se netko može zapitati što bi ostalo od Leonardove slave da je svoje crteže radio na isti način kao što tinejdžeri ubijaju marsijance u video-igrama. Crtanje pomoću tastature je sasvim mazohistički podvig i mnogo bolja alternativa je grafički tablet (digitizing pad) po kojem se crta pomoću pisala koja bi bilo



Kompjutor nudi umjetnicima potpuno nove postupke i materijale. Kad se osnovni crtež unese u memoriju kompjutora, može se mijenjati na tisuću načina: poboljšavati, neke boje isticati, a druge zapostavljati i zasjenjivati.

Desno Slike koje su otisnute na početku svakog dijela ove knjige napravljene su na likovnom video-sustavu FLAIR, na kojemu se crta pomoću grafičkog tableta s elektroničkim pisalom. Na donjem dijelu tableta nalazi se područje preko kojega se upravlja različitim grafičkim funkcijama, kao: izaberi boju (između 255 mogućih boja koje su prikazane na ekranu) ili izaberi oblik (ili veličinu, debljinu crte itd.), nacrtaj kružnicu, elipsu ... Posebni softver osigurava najrazličitija modificiranja slike. Ovdje u okviru unutar slike crtač je povećao jedan detalj kako bi ga mogao preciznije obraditi. Paleta boja vidi se na dnu ekrana. Različite faze slike pohranjuju se na disketu s koje se mogu uvijek pozvati. FLARE se koristi i kao generator teksta (character generator) visoke rezolucije i s velikim mogućnostima komponiranja natpisa na ekranu.



sasvim slično olovci, da nije žicom vezano za tablet.

Nakon što su kontura spremljene u memoriju može se pristupiti bojenju. Kursor treba samo dovesti unutar neke plohe i narediti da je oboji grimizno sa zeleno-smeđim mrljama. Ako to nije lijepo (kao što i nije), boje se mogu smjesta promijeniti, recimo u narančastu sa sivoljubičastim mrljama.

Ipak, ovo samo manjim dijelom iskorištava potencijal računala. Dobar grafički program morao bi raspolagati bibliotekom 'pod-slika' koje bi se mogle dodavati glavnoj slici u različitim položajima, veličini i boji. Bilo bi to vrlo korisno, na primjer arhitektima, koji bi perspektivne skice svojih projekata mogli jednostavno dopunjavati realističkim detaljima okoliša.

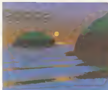
Nije nužno biti vješt crtač, dovoljno je znati na

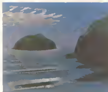




pravi način upotrijebiti TV kameru, grafički tablet i kompjutor. Kao podloga za sliku njujorske podzemne željeznice na ovoj stranici vjerojatno je poslužila fotografija. Pomoću kompjutora može se mijenjati njezina struktura, perspektiva i boje. Može se miješati nekoliko slika, ponavljati oblike i sve u svemu — toliko promijeniti osnovni crtež da ga ni vlastiti tvorac ne bi prepoznao.

Nije teško zamisliti kako će za nekoliko godina, kada ovakvi sistemi budu u svakodnevnoj uporabi, trgovine umjetnina prodavati digitalizirane verzije klasičnih umjetničkih djela. Tako ćete valjda moći kupiti Monu Lisu ili Ledu s labudom na disketi i kod kuće ih pretvarati u najsmjelije likovne fantazije.







Karlin otok (Carla's Island) je prvi kompjutorom generirani film u kojem se svi pokretni elementi (sunce mjesec, valovi i oblaci) ponašaju kao stvarni fizički objekti. Pogledajte kako se svjetlo sunca na zalazu reflektira na stražnjim dijelovima valova dok se u ostalom dijelu vode reflektira nebo, kako oblaci mijenjaju boju dok se mjesec pojavljuje za njih i onaj nježni odsjaj mjesečine na obalinama brda. Precizno je određen put svake zrake svjetla u slici — da li će biti raspršena, reflektirana ili apsorbirana kada se dotakne određena površine — sve to izračunava dio programa kojim se simuliraju čvrsti objekti. To zahtijeva mnogo računanja, jer za svaku dvadesetpetinku sekunde trajanja filma treba po nekoliko minuta računanja na CRAY kompjutoru, najvećem i najmoćnijem na svijetu.

U sredini: Panza, industrijski robot, konstruiran za eksperimentalni film Ra-dovi (The Works), proizveden u njujorškom Tehnološkom institutu na VAX kompjutoru i na uređaju za filmsko snimanje Dicho-med D48.



Kompjutor može lako crtati kuća i druge objekte, ali bi oni izgledali kao da su prozirni. Da bi se izbjegle skrivene linije, kompjutor mora izračunati udaljenost od zamišljene pozicije kompjutorova oka do najbliže plohe i sve što je skriveno iza takvih ploha ne smije se vidjeti. Budući da se operacija ponavlja na svim objektima što se nalaze na slici, potrebno je mnogo računati te su zato trodimenzionalne animacije (popuf one na str. 106) vrlo komplicirani kompjutorski zadaci.

Jedan od najdosadnijih poslova je crtanje animiranih filmova. Za jednu minutu gotova filma potrebno je izraditi 1500 crteža (brojka nije sasvim točna, jer se obično za jednu minutu filma izrađuje dva do tri puta manje crteža — *op. prev.*), a za čitav film i nekoliko stotina tisuća. Gledati filmske animatore kako rade isto je što i promatrati travu kako raste. Zbog toga nas ne iznenađuje da su kompjutori, čim su postali dovoljno moćni, odmah upotrijebljeni za izradu crtanih filmova.

U studijima crtanog filma najbolji crtači izrađuju samo krajnje faze pokreta — Tom podiže nogu ili Tom udara Jerryja — a njihovi pomoćnici izrađuju međufaze, crtaju Tomovu nogu kako se malo po malo približava jednom Jerryju. Potpuno je razumljivo da se za ovaj dio posla upotrijebi kompjutor.

Pretpostavimo da u sceni vidimo buldoga gdje mu je u njušku iznenada uletjela cijev za napuhavanje automobilskih guma. Umjesto da se crta slika po slika buldoga kako se napuhava, treba nacrtati samo početnu i završnu sliku, a sve ostalo će obaviti animacijski kompjutorski sistem koji će po zadanom programu izračunati proporcionalne točke i prema njima izraditi određene crteže.

Da uštede rad i sačuvaju kontinuitet, filmski animatori nikada ne crtaju cijelu sliku. Lica koja se kreću u prednjem planu crtaju se na listovima prozirne plastike, dok se pozadina crtaju posebno i koriste nekoliko puta. Pozadina može biti pokretna, što zahtijeva perfektno usklađivanja pokreta u prednjem planu i kretanja pozadine, pa je za kompjutor upravo idealni zadatak da upravlja ovom fazom posla. Danas već postoje i kućni kompjutori koji mogu praviti animirane filmove, ali dakako s mnogo nižom rezolucijom od profesionalnih sistema.

Kompjutor može uvelike pomoći u bojanju crteža. Njime se mogu brzo i jednostavno mijenjati boje, pa crtač može iskušati mnogo više kombinacija nego na papiru.

Trodimenzionalne grafike

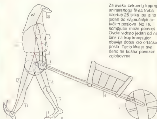
Kompjutor je na svojem pravom terenu kada treba izraditi trodimenzionalne prikaze objekata, što je daleko teže i zanimljivije od pripremanja dvodimenzionalnih crteža. Kada su u pitanju komplicirani proračuni potrebni za crteže u perspektivi, nema idealnijeg crtačeva pomoćnika od kompjutora.

Da se neki predmet prikaže u perspektivi, sve horizontalne linije moraju voditi iz jedne ishodišne točke (očišta) koja se može nalaziti na bilo kojem mjestu horizonta. Kompjutoru nije teško



da iz zadanih koordinata izračuna sve točke kuće (na slici lijevo) gledana iz bilo koje perspektive. Problem je jedino u tome što će tako nacrtana kuća biti providna, ali i to nije nerješivo, mada uklanjanje skrivenih linija traži mnogo strpljiva programerskoga rada.

Kuća, sa svojim ravnim plohama, nije najtipičniji oblik u prirodi. Većina objekata ima zakrivljene, često vrlo komplicirane plohe. Sastavljači programa za trodimenzionalnu grafiku moraju birati između dviju mogućnosti: izračunati koordinate svake točke objekta i pohraniti ih u memoriju ili rastaviti sliku na mnogo malih dijelova pa svaki posebno pohraniti. Pokazalo se da je prva varijanta jako skupa i da je mnogo bolje — premda i to zahtijeva dosta vremena i truda — pretvoriti svaku zakrivljenu površinu u ravne poligone pa njih pohraniti u memoriju. I ovaj postupak ima, dakako, svojih mana, jer na primjer ni jedan Harrier ne izgleda poput onoga na stranici 106. Bila bi potrebna dalja softverska obrada da bi se izgladili poligoni i izgubio fasetirani oblik zakrivljenih površina. Ali takvi programi troše mnogo vremena i s njima se ne bi mogla ostvariti prirodna (real time) animacija.



Za svaku sekundu trajanja animiranoga filma treba nacrtati 25 slika, pa je to jedan od najmučnijih crtačkih poslova. No i tu kompjutor može pomoći. Ovdje vidimo jedan od načina na koji kompjutor obavlja dobar dio crtačkog posla. Tijelo lika je svedeno na kostur povezan zglobovima.



Da bi se lik pomicao, potrebno je nacrtati samo dijelove kostura, što sve kako traži mnogo manje vremena nego crtanje cijelog lika. Ako je potrebno, pojedini dijelovi tijela mogu se kompjutorski mesti ili rastezati, što ovisi o vrsti pokreta. Dijelovi u drugom planu prekriveni su onima što se vide u prednjem planu. Kolač se može okretati naredbom SPIN.



Ljudsko tijelo se giba oko jedne točke koja se nalazi u predjelu struka, a trup i noge se savijaju naprijed ili natrag. Mora se samo grubo naznačiti pokret, a kompjutor nastavlja crtanje, izračunava i dorađuje oblike, kolorira sličicu po sličicu, jednom riječi — in stant animirani film.



Tri slike s lijeve strane kompjutorske grafike otvara sasvim nove prostore za kreativnost. Ovdje vidimo fantastično približavanje gradu iz mašte koje se dojma poput čudesnog leta helikopterom

Tri slike s desne strane kompjutorskom grafikom se danas često služi televizija u reklamne svrhe da bi se oko reklamiranog proizvoda stvorio nimbus visoke tehnologije. Ova obična četkica za zube djeluje kao neki čudni alat iz arsenala znanstvene fantastike

Na suprotnoj stranici usavršeni grafički kompjutori oslobađaju maštu umjetnike. Ova gruda spiralnih opruga kao da je izašla iz Escherove grafičke radionice vjerojatno je generirana pomoću vrlo jednostavnog algoritma a rezultat mora da je iznenadio i samog autora

Slika se ne sastoji samo od čistih oblika. Na objektima u prirodi vide se svjetlo i sjena, boja i odrazi drugih objekata. Posebno je važna njihova struktura, jer površina opeke, na primjer, potpuno je drugačija od površine stakla. Da bi se postigao savršeni efekt trodimenzionalne grafike, kompjutor mora, uzevši u obzir smjer svjetla, izračunati reflektiranje od različitih površina i raspršivanje svjetla kao posljedicu reflektiranja. Osim toga, slike mogu biti pokretne. Kompjutor bi morao biti programiran tako da izračuna pokrete u slici — kretanje valova, na primjer, mora imati svoju težinu, mora se vladati prema svim fizikalnim zakonima. Svaki čovjek se kreće na neki određeni način pa i to kompjutor mora izračunati. Zato ne začuđuje što tako komplicirane slike zahtijevaju nekoliko sati obrade i na najmoćnijim kompjutorima

Posljedica toga je da kompjutori sudjeluju u umjetničkom stvaranju na sve višoj razini. Na početku su igrali ulogu nespretnih pomagača, obavljali pomoćne poslove: bojili drveće zelenom, a nebo plavom bojom. Danas već mogu preuzeti i dobar dio crtačeva posla. Slikar će, recimo, zadati elemente nekog pejzaža samo u glavnim crtama: planina tamo, dolina ovdje, a malo dalje jezero. Stroj će tada, imajući na umu (u programu) zakone geologije, organizirati konture pejzaža i uzevši u obzir godišnje doba opskrbiti ga prikladnom vegetacijom i bojom i osvijetliti svjetlom koje najbolje odgovara dobu dana. I — što još preostaje umjetniku? Vjerojatno ono najvrednije: zamisliti cijelu stvar

GLEDANJE KOMPJUTOROM

Da bi se ilustrirao niz slika (na suprotnoj stranici) koji prikazuje proces digitalizacije, ispred TV kamere stavljena je fotografija ovog ruševnog dvorca. Slika je rastavljena na piksele i svjetlosni intenzitet svakog piksela pohranjen u memoriju





Ovih osam fotografija predstavlja faze digitalizacije procesa kojim se dolazi do osnovnih podataka što služe za potpuno kompjutorsko gledanje

Na stranicama 108-109 vidjeli smo da se slika može rastaviti na piksele i pohraniti u kompjutorsku memoriju. Postići međutim da kompjutor progleda, da od memoriranih podataka sastavi upotrebljivu sliku, nije nimalo lako, što dokazuje da je ljudski mozak još uvijek mnogo pametniji od najmoćnijeg kompjutora.

Pri analizi slike električki šum je prva poteškoća s kojom se treba suočiti. Odstupanja će se pojaviti čak i onda ako kameru usmjerimo na sasvim jednoličnu sivu plohu, koja bi trebala napuniti memoriju pikselima od kojih svaki ima, recimo, vrijednost 127. Srednja vrijednost iznosit će 127, ali pojedine ćelije će fluktuirati oko te vrijednosti. Bit će ih od 126 i 128, pa čak i od 130 i 124. Odstupanja uzrokuju sitne nestabilnosti u kameri, pojačalima i analogno-digitalnom pretvaraču. Da bi se suzbio šum, treba najprije usporediti svaki piksel s njegovih osam neposrednih susjeda, pa ako se neki od njih previše razlikuje od susjednoga, treba ga zamijeniti s prosječnom vrijednosti.

Čovjek percipira sliku okom i umom služeći se pri tome s nekoliko različitih procesa. Najprije raspoznaje konture objekata, zatim nalazi područje sličnoga tona, služi se sjenom kao znakom za detekciju volumena, a ispred kompjutora ima prednost što usporedbom dviju sasvim neznatno drugačijih slika postiže stereoskopsku iluziju prostora.

Da bi shvatio ono što treba „vidjeti" kompjutor mora poći od analize kontura, odnosno svesti sliku na crtež osnovnih oblika, što se postiže tako da se usporedbom svakog piksela nađu oni koji se znatno razlikuju od prosječnih vrijednosti, pa se ti pikseli zadrže, a ostali odbacuju. Rezultat je (ili bi bar morao biti) konturni crtež objekta. Svrha ovog postupka je stvaranje zatvorenih petlji, jer svaki čvrsti objekt ima konturu koja ga sa svih strana okružuje završavajući uvijek u početnoj točki.

Na žalost, u stvarnosti nije tako. Konture se prekidaju i gube, pogotovo kad kompjutor postupkom određivanja srednje vrijednosti ne može uočiti dovoljno velike razlike između susjednih piksela, ili kad odrazi stvaraju nesuvisle promjene u intenzitetu. Sjene su također veliki problem, jer ih nije moguće analizirati bez poznavanja položaja izvora svjetla. Poteškoćama treba još pribrojiti i zabune koje se javljaju kad bliži objekti djelomično prekrivaju one udaljenije. Ponešto može pomoći tehnika nazvana „vertex analiza", koja koristi činjenicu da većina objekata što ih kompjutor „gleda" ima ravne rubove i kutove. Analizom kutova kompjutor može odijeliti realne kutove od onih koji su nastali međusobnim preklapanjem objekata i do izvjesne mjere ispuniti praznine u konturama, te razdvojiti jedan objekt od drugog.

Raspoznavanje slike može se također postići tehnikom nalaženja sličnih tonova. Metodom slučajnog odabiranja uzima se jedan piksel za uzorak i provjerava da li su susjedni dovoljno slična intenziteta. Ako jesu, procesor ih ozna-

Na slici A vidi se rezolucija pri rastavljanju slike na 2 × 2 piksela. Slika B ima 8 × 8, a slika C 32 × 32 piksela. Na slici D ima 256 × 256 piksela, što iznosi 64K, a svaki se može prikazati u 256 razina sivoga, tako da svaki zauzima po jedan bajt memorijskog prostora. Standardna TV rezolucija iznosi oko 600 × 600 piksela, pa za memoriranje crno/bijele slike treba 360 000 bajtova, a za sliku u boji preko pola milijuna.

Na slikama E i F prikazan je postupak određivanja kontura. Slika F je povećani sadržaj malog bijelog okvira na dnu slike E, gdje se vide promjene svjetlosnog intenziteta do kojih dolazi između zida dvorca i trave. Usporedbom vrijednosti kompjutor deducira tok konture.

Na slici G vidimo konture cijele slike. Rezultat je čak i u ovako jednostavnom predlošku prilično zbrkan.

Druga tehnika kojom se služi kompjutor da bi razaznao sadržaj slike je određivanje područja sličnoga tona, a sastoji se u grupiranju piksela čija je svjetlosna razina slična razini susjednih piksela. Rezultat takve analize vidi se na slici H.

Idući korak je pokušaj da se isprekidane konturne linije sastave u suvislu konturu objekta. Tada stupa u akciju program koji ispituje kutove i uglove (vertex analiza — vidi str. 118—119) i pokušava steći pojam o trodimenzionalnosti objekta. Nakon toga može se vratiti na početak i pozabaviti se sjenama koje bi se inače mogle interpretirati kao objekti.
Dok ljudsko oko shvaća sliku u jednom trenutku, kompjutorski sustav može obrađivati samo relativno jednostavne slike, što u laboratorijskim uvjetima može trajati minutama ili i satima, čak i na vrlo moćnim kompjutorima. Pravo gledanje kompjutorom bit će moguće tek kada se pojave daleko moćniji strojevi s paralelnim procesorima. (Vidi str. 174—175.)

čeve te ne tej način 'rastu' područja sličnog intenziteta i kompjutor pretpostavlja da su to dijelovi jedne cjeline Kad program ne nalazi više jednakovrijednih piksela odabire novu polaznu točku i pokušava odrediti novo područje sličnog tona Kombinacijom ovog postupka i 'vertex analiza' program će pokušati uspostaviti realne odnose između područja sličnoga tona i kontura i tako ispuniti praznine u logici slike

Ljudska optika služi se i mnogim fizikalnim zakonima za razvrstavanje i razaznevanje viđenoga Čovjek uvijek ima prilično točnu predodžbu o debljini zidova visini kuća ili udaljenosti planina Objekti koji ne odgovaraju ovim predodžbama bilo da su pretanki previsoki ili predebeli djeluju čudno stvaraju zabunu i čovjek pretpostavlja da su rezultati optičke varke — pogrešne analize Ako se objekti kreću čovjek zna da se obično kreću pravocrtno e ako zakreću onda će to činiti u blagim lukovima Kad u dubini drvoreda ugledamo svijetlu mrlju iznad okomite tamne mrlje pretpostavit ćemo da se radi o ženi u tamnoj haljini sa svijetlim šeširom na glavi Ali ako se mrlje najednom rastave ili se stanu kretati različitim brzinama bit ćemo sasvim zbunjeni

Čovjek u svojoj memoriji čuva ogromnu zalihu uskladištenih prizora On dobro zna kako izgledaju ljudi a kako stolovi rafinerije nafte cvijeće ili insekti Stalno uspoređuje ono što vidi s onim što je uskladišteno u njegovoj memoriji pa kad klikne gledajući u daljinu To je ptica pe malo zatim Ne to je avion „Ne Ne! To je supermen!" znači da objekt koji mu se približava ponovo uspoređuje sa slikama iz svoje memorije preinačuje procjene i stvara nove zaključke

Na žalost kompjutor je još vrlo daleko od ovako savršene obrade podataka Svaki put kad nešto gleda počinje iznova kao da nikada ranije nije ništa vidio

Veliko područje primjene kompjutorskog gledanja' nalazi se u industriji a posebno u rudnicima i unutrašnjosti nuklearnih reaktora

Prividno jednostavna zadaća identifikacije pojedinih dijelova koje treba ugraditi u neki stroj ipak nije sasvim jednostavna Kompjutor se oslanja samo na dvodimenzionalnu sliku i to je jedna od najvećih smetnji koju treba savladati Stoga se dijelovi da bi se što jasnije ocrtavale njihove konture stavljaju na osvijetljenu podlogu ispred televizijske kamere koja predstavlja kompjutorovo oko Ima dijelova koji su plosnati (na primjer zupčanici) pa s njima nema problema ali ima ih i takvih koje kompjutor može prepoznati samo ako se nalaze u unaprijed točno određenom položaju

Kada kompjutor dobije konturu predmeta nastoji ga prepoznati na različite načine Najprije mjeri njegovu veličinu pa iz toga izračunava određeni broj parametara i zaključuje oblik Nedavno je u Zavodu za umjetnu inteligenciju Sveučilišta u Edinburghu realiziran program koji je prepoznavao čokoladne bombone (izabrane zbog jednostavnih oblika s finim razlikama) Kompjutor je najprije mjerio njihovu površinu zatim opseg maksimalnu i minimalnu širinu itd Stroju su pokazani primjerci čokoladnih bombona s karamelom višnjom rumom maslacem narenčom i konjakom e stručni sustav (expert system — vidi str 86) je izradio pravilo za identifikaciju različitih okusa služeći se mjerenjem različitih parametara ponuđenih bombona

Ljudsko oko lako razpoznaje osobu koja hoda i prema pokretima zaključuje veličinu, oblik i poziciju udova Čini se da je to jednostavan zadatak ali kompjutoru za isti proces treba nekoliko sati Na slici vidimo kako program napisan na Sveučilištu Sussex zamjenjuje trup i udove čovjeka jednostavnim valjkastim oblicima za dokaz da je na ispravan način analizirao njegova kretnje

Lijevo sustavi za kompjutorsko gledanje imaju najveću primjenu u tvornicama gdje se predmeti koje moraju vidjeti redovito sastoje od ravnih linija i uglova. Na slici vidimo jedan objekt sa svim mogućim uglovima: konveksnim, konkavnim i okluzivnim. Kompjutor ispituje dvodimenzionalnu sliku određujući uglove u skladu s mogućim tipovima. Nakon što je obišao cijeli oblik, prvi kut mora biti u skladu sa zadnjim. Ako se kutovi ne slažu, objekt je nemoguć, poput đavolje viljuške koju vidimo na slici gore.

Dolje lijevo: jednostavan način da se kompjutor nauči prepoznavati mehanička dijelove koji mogu ležati u različitim položajima sastoji se u tome da se svaki dio stavi na svijetlu plohu u nekoliko karakterističnih položaja da bi ga kompjutor kasnije mogao prepoznati u različitim projekcijama.

Na dnu lijevo: da bi razlučio oblike, rad kompjutora se mora svesti na što jednostavnija mjerenja. Ovdje vidimo kako stroj razaznaje čokoladne bombone pomoću programa koji je razvijen na edinburškom Sveučilištu. Mjeri dužinu, širinu i promjer kružnice koja se može opisati oko predmeta i upisati u njega.

Dolje: slika predmeta projicira se na digitalizator. Svaka ćelija javlja intenzitet svjetla koje na nju pada, a kompjutor određuje rubove objekta uspoređujući intenzitete svjetla pojedinih ćelija u odnosu prema susjednim. Ako je razlika velika, zaključuje da se na tom mjestu nalazi rub predmeta.

KOMPJUTOR KOJI GOVORI

Zamišena slika zvučnih valova govore primljena mikrofonom i snimljena sa ekrana osciloskopa. Kompjutori koji bi trebali razumijevati ljudski govor moraju rastaviti ovakve zvučne valove u različite frekventne opsege i tada usporediti pohranjene vrijednosti s riječima koje već znaju



Postoji u kompjutorskom svijetu jedna struja koja smatra da kompjutori neće pravo koristiti sve dok ne budu mogli govoriti i razumijevati govor. Prvo je danas već sasvim moguće, ali drugo, kao što ćemo vidjeti, nije tako jednostavno. Najprimjerenija upotreba kompjutora koji govori je u situacijama kada korisnici ne mogu čitati, jer su ili slijepi ili su previše zauzeti drugim poslovima (kao na primjer u avijaciji ili automobilizmu), pa ne mogu skretati pogled sa svog glavnog zadatka.

Ima dva načina da se izvede digitalno zapisivanje i reprodukcija govora. Prvi način je da se promjenljiva naponska razine stvorene u mikrofonu diskretiziraju na po prilici 8 kHz, pretvore u digitalne brojeve i pohrane u memoriju kao serije bajtove. To je u principu moguće, ali za to treba oko 8 kilobajta memorije za svaku sekundu govora. Dakle, na disketu kapaciteta 5 megabajta moglo bi se pohraniti samo 10 minuta govora.

To očito nije previše praktično i može se dakako i bolje riješiti, ali najprije treba pažljivo proučiti ljudski govorni mehanizam. Ljudski glasovni trakt sastoji se od ustiju, koja su neka vrst cijevi promjenljive dužine, što se sve više sužava prema grlu. Jezik i mišići u grlu mijenjaju dužinu ove cijevi, a posljedica toga je promjenljiva rezonantne frekvencija.

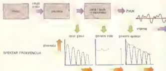
Suglasnici ponajviše zvuče kao praskovi i psikanja, što ovisi o vibracijama zračnoga stupa. Da bi se to imitiralo, treba samo napraviti izvor 'bijelog šuma', to jest šuma u kojem su podjednako zastupane sve slušne frekvencije, što se elektronički lako postiže jednostavnim pojačavanjem napona na izlazu Zenerove diode.

Glasnice u grlu, zapravo kožnati nabori, mogu se zatezati i opuštati tako da titraju na različitim frekvencijama dok zrak iz pluća struji preko njih, a psikanje se oblikuje u ustima strujanjem zraka između zubi i jezika. Glasnice titraju na različitim frekvencijama što je ovisno o promjenama veličine i oblika usne šupljine.

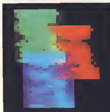
U elektronskoj verziji ulogu glasnica preuzima generator frekvencija, a promjenljivu rezonantnu šupljinu ustiju zamjenjuju tri digitalno kontrolirana filtera. Tako je riješeno pitanje samoglasnika, a riječi kao daj i roj koje zahtijevaju vrlo brzu promjenu frekvencije formanata, što se u ustima izvodi pomoću kretanja jezika, elektronski se oponašaju mijenjanjem frekvencije formanata usred samoglasnika.

Postoje još i nazalni glasovi m, ng i n koji nastaju karakterističnom rezonancom nosa od oko 1400 Hz, što se u kompjutoru rješava ugradnjom posebnog filtera, i konačno frikativi (trveni glasovi) — s, š, z, ž, th i f što ih stvaraju jezik i zubi bez udjela rezonance glasovnog trakta. Ako tome dodamo još i raznolika kucanja, zviždanja, psikanja i cvoktanja kojima obiluje naš govor i podsjetimo se na stih „... cvrči cvrči cvrčak na čvoru crne smrče", onda postaje jasno koliko problema stoji na putu pretvaranja teksta u govorni jezik. Dugotrajnim učenjem

Dolje desno: govor se može razbiti u različite frekventne opsege i kompjutori tada prepoznaju oblik svakog glasa. Ovdje se vidi kako izgleda analiza riječi zero. U praksi taj sustav može prepoznati nekih stotinjak riječi, ali samo ako ih izgovara osoba posebno za to izvježbana



Ljudski je govor vrlo kompliciran. Naše glasnice proizvode različite visine tonova s frekvencijama koje rastu i intenzitetom koji opada. Usne šupljine i jezik tvore cijev promjenljive veličine poput one na orguljama, instrument kojim se neke frekvencije mogu posebno isticati (formanti). Sasvim desno vidi se rezultat analize: plavim i zelenim su prikazane niske frekvencije, a crvenim visoke





pomoću naredbi koje živčani sustav šalje u mišićje grla, čovjek taj proces obavlja naoko jednostavno, ali mnogo je kompliciranije kada isti put treba oponašati kompjutor. Čovjek obično govori brzinom od oko 100 riječi u minuti (prosječna riječ ima oko 6 slova) i govor kodiran ovim načinom zahtijeva oko 10 bajtova u sekundi — to jest 600 podataka potrebnih za direktno digitalizirani govor, što predstavlja znatnu uštedu memorijskog prostora.

Danas za tu vrst posla postoji nekoliko čipova. Jedan od njih, SC-01, nudi suglasnike f g k t z b d j, p h m n l r w th sh i ch i samoglasnike a ah ae aw e eh i o oo u uh i y koji dolaze u dvije ili više dužina što se označavaju indeksom. Ima na primjer 4 vrste glasa e u trajanju od 59 do 185 milisekundi i sa stankama također različite dužine. Ako se SC-01 napuni normalno napisanim engleskim tekstom rezultat nije upotrebljiv. Da bi se čula riječ "kompjutor" treba upisati K UH_3 M P U_1 T ER, a za frazu "good afternoon" (dobar dan) treba napisati G OO D AH_1 F T UH_3 NU U_1 N. Kao što se vidi, priprema teksta za kompjutorski govor nije nimalo jednostavna.

Posebni problem pretstavljaju različiti lokalni izgovori, jer ono što dobro zvuči Teksašaninu, Škotu će biti jedva razumljivo (igračka "Speak and Spell" tvrtke Texas Instruments, koja se služi nešto drugačijom tehnologijom, nije imala uspjeha u Britaniji, jer nitko nije dobro razumio njezin američki izgovor).

Ima još jedan problem, a to je što zapravo često i nije potrebno jasno čuti neku riječ da bi se shvatio njezin smisao, već je dovoljno osjetiti intonaciju i naglasak. Kompjutor bi stoga morao "razumjeti" općeniti smisao govora, da li zvuči agresivno ili uvjeravajući, ljubazno ili dosadno, te u pravom trenutku izabrati pravu intonaciju. To dakako još nije riješeno, a čini se da se do sada još nitko tim problemom nije počeo ozbiljno baviti.

Glazbeni softver

Iako ljudski govor ima neke posebnosti s kojima kompjutor teško izlazi na kraj, kada se radi o glazbi onda upravo kompjutor može biti idealan pomoćnik kompozitoru.

Zna se kako je zapisivanje nota, transponiranje tonaliteta i razrada partiture mučan posao. Danas ima već nekoliko kompjutorskih sistema koji su posebno za to konstruirani. Jedan od takvih je Synclavier opremljen klavijaturom na kojoj se može svirati kao na svakom sintesajzeru. Svaka odsvirana nota pojavljuje se istodobno na ekranu ispisana na notnom crtovlju. Istodobno se mogu ispisivati note za 16 instrumenata, a odsvirana i zapisana glazba pohranjuje se na disketu. Zvuk svakog instrumenta može se prekrajati po miloj volji uz pomoć Fourierove analize (vidi str. 108—109).



Ima mnogo sličnosti između kompjutorskog programa i glazbe. Glazbeni zapis sastoji se od mnoštva potprograma kojima je specificirano trajanje, harmonija, tonalitet itd. Isti glazbeni motiv može se javljati više puta i na različite načine, pa nema razloga da ga kompozitor ponovo zapisuje. Ako ga ipak zatreba, može ga pozvati kad god zaželi u obliku potprograma, ali sa različitim parametrima, što mu sve može pojednostavniti posao.

Synclavier može obaviti i sinkronizaciju glazbe na filmu ili videu. Kompozitor treba samo odrediti točan broj faza koje zahtijeva medij u kojem radi i ugraditi sinkrone oznake u kompjutorsku partituru. Sve ostalo obavit će kompjutor.

Nije neobično što za sav taj posao treba stroj sa 16-bitnim procesorom u vrlo skupoj konfiguraciji. No kada se uzme u obzir ogromna produkcija glazbe posebno pisane za film, televiziju i pop sastave, onda cijena ne igra nikakvu ulogu, pogotovo ako se usporedi sa olakšicama koje pruža kompjutor.



KOMPJUTOR KOJI ČUJE

Vidjeli smo kako se kompjutor može naučiti da govori iako to još nikome nije baš najbolje uspjelo. Ali navesti kompjutor da "čuje" još je mnogo, mnogo teže. Ne zna se točno koliko je to teško jer još nitko nije postigao neki zadovoljavajući rezultat.

Nama je naš jezik jednostavan pa smo skloni potcijenjivati probleme što ih on nameće stroju. Problemi su slojeviti i međusobno uvjetovani. Ponajprije treba onu nejasnu i nesistematičnu buku koju proizvode naša usta pretvoriti u relativno jasan slijed glasova. Ako ste ikada pokušali transkribirati magnetofonsku snimku nekog razgovora, bit će vam jasno kako je teško razaznati što ljudi govore, premda sudionici u razgovoru nisu imali nikakvih taškoća u sporazumijevanju. Naš je govor pun zastajanja i nerazgovjetnih poštapalica poput "eee", "aaa", "mmm", "ovaj" i "jelda". Riječi koje bi napisane trebale biti razdvojene razmacima izgovorene su zajedno, a razmaci se pojavljuju usred riječi na sasvim neočekivanim mjestima. Da bi se potpuno razumio tekst, treba znati temu razgovora pa se tek tada mogu razlikovati riječi koje su istoga zvuka a različitih značenja, kao u slučaju "marka" i "Marka" (genitiv od Marko). Sve to dokazuje da pravi smisao riječi ovisi o kontekstu i da nije dovoljno samo uhom slijediti nečije misli. Kompjutor koji treba razumijevati ljudski govor morao bi u prvom redu shvaćati o čemu se govori, a to je najveća zapreka.

Razumijevanje je jedan od najtežih problema današnjice i jasno je da nije dovoljno razumjeti samo riječi da bi se mogle slijediti nečije misli. Kada netko kaže "danas je bila tuča", tko će znati da li se radi o fizičkom obračunu ili o meteorološkoj pojavi. Ili kad netko kaže da li su pospremili stvari u "kako se ono zove" kao i zadnji put, nema nade da ćete razumjeti ukoliko niste obratili pažnju na ono što je bilo rečeno ranije. Često netko upre prstom i kaže "onaj tamo ne, ne, ne ovaj, nego onaj". Gesta ruke je ta koja nam je pomogla da shvatimo smisao, a kompjutor još uvijek nije u stanju da odgoneta smisao geste.

Ne radi se dakako samo o optičkom kontaktu sa sugovornikom, već o angažiranju cjelokupnog znanja, iskustva i poznavanja prošlih događaja. Neka od tih znanja su toliko općenita da su svima poznata. Primjerice kad se u meteorološkom izvještaju govori o tuči, svakoma ća biti jasno da se ne radi o fizičkom obračunu.

Čak kad bi kompjutor imao dovoljno predznanja za razumljevanje razgovora, još uvijek se ne zna na koji bi način stroj iskušao sva moguća značenja nekoga pojma. Ono za čim se teži je razvijena struktura obrade podataka koja dostavlja sve moguće interpretacije zvukova procesoru na višem nivou, a ovaj pak stvara sliku razgovora i prihvaća ili odbacuje interpretaciju u skladu s onim što je već ranije o toj temi izrečeno.

Do danas je najdalje došao sistem, čiji je prototip razvijen u Bellovim laboratorijima, koji preko telefona daje obavijesti o rezervacijama za zrakoplovne linije.

Taj sistem ne treba ništa znati o svijetu izvan zrakoplovnih redova letenja i poznavati strukturu jednostavnih engleskih rečenica. Kada mu glas s druge strane žice kaže "u subotu bih želio letjeti u San Francisco", ne treba mnogo različitih znanja da bi se shvatilo o čemu je riječ.

Sistem osluškuje stanke između riječi, dijeli govor na riječi, a riječi na odsječke duga 15 milisekundi, u kojima frekvencija i intenzitet zvuka ostaju približno konstantni. Zvukovi se tada dijele na frekventne opsege širine jedne oktave — 100-200, 200-400, 400-800, 800-1600 i 1600-3200 Hz — i kompjutor pamti frekventni opseg i intenzitet zvuka svakog odsječka. Pošto je na taj način reducirao svaku riječ na skup brojeva, može pokušati da je usporedi s riječima koje se već nalaze u njegovoj memoriji. Te riječi ne moraju biti identične s riječima iz memorije, ali kako za razgovor o redu letenja nije potreban





Kao dio istraživanja ljudskoga mozga na Brunel sveučilištu u Londonu sagradili su stroj koji čini se može u nekim aspektima imitirati živčani sustav. Digitalizirana TV slika lica s vlastitim vezama među pikselima, može se naučiti da raspoznaje različite ljude i izraze njihovih lica — nasmijanih ili namrgođenih

neki bogat rječnik nece biti previse kandidata za svaku riječ

Pošto je pronašao sve kandidate za neku riječ stroj pokušava svakog posebno uklopiti u rečeničnu strukturu Dijelovi rečenice u subotu i u San Francisco odmah će biti jasni, ali ono između može zbuniti stroj Ali kako ne može biti govora ni o čemu drugom osim o letenju čak ni stroju nece biti teško da zaključi o čemu se radi i da samostalno odustane od daljeg pretraživanja memorije

Kada se iste tehnologije primijeni na općenitije sadržaje iskrsavaju mnogi novi problemi teži nego što su eksplozije mogućih značenja nepoznatih riječi Da bi se ograničilo beskonačno pretraživanje memoriranog rječnika sistem koristi gramatička pravila koja govorniku sluze za povezivanje riječi u rečenici u cjeloviti smisao. Taj način djeluje dobro dok se radi o sasvim jednostavnim rečenicama kao na primjer Crvenkapica (subjekt) ima (predikat) košaricu (objekt) Nevolja je sto se u praksi takve jednostavne rečenice rijetko susrecu

Što se dublje ulazi u jezičnu problematiku sve se više čini da svaka riječ ima svoj vlastit skup gramatičkih pravila prema kojima se ponaša što uveliko komplicira cijeli problem Ne može se jednostavno zaključiti slijedeća riječ mora da je glagol pa ću u memoriji tražiti samo glagole jer ce se uvijek naći dvije-tri imenice koje se točno uklapaju u rečeničnu strukturu

Problemi su tako složeni da je već mnogo programersko srce prepuklo pod teretom zadatka a kada idući put na filmu vidite čovjekolika robota kako govori tečno iako monotonim glasom znajte da on ipak spada u znanstvenu fantastiku

Kompjutori surađuju s vanjskim svijetom na mnogo načina i se mnogo raznih naprave

Desno oblik modela prenosi se u stroj rukom za digitalizaciju čiji vršak točno slijedi oblik modela. Uređaj za precizno mjerenje kutova nalazi se u zglobovima naprave

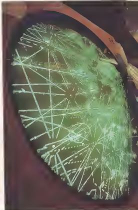
Dolje desno kompjutoru je svejedno što i za koga radi. Lakoća s kojom mjeri količinu tekućine pomoću senzora za razinu i način na koji upravlja crpkama i ventilima čine ga idealnim za ovu vrst poslova. Na slici vidimo kontrolnu dvoranu vodovodne mreže a dolje u sredini postrojenje pivovare.

U sredini kao senzori mogu služiti radari kilometrima udaljeni od kompjutora. Na slici vidimo kontrolore zračnog prometa nad kompjutorskom slikom zračnog prostora od 50 000 kvadratnih kilometara. S palicom (joystick) u rukama kontrolor može jednim pokretom skrenuti s puta i najbrži mlažnjak

Sasvim desno, satelitska antena povezuje ured kompanije s platformom podmorske bušotine udaljene nekoliko stotina kilometara. Udaljena mjesta poput ovog bit će povezana preko satelita a ona bliža u gusto naseljenim područjima održavat će veze putem vodiča od optičkih vlakana







Ako kompjutori uopće mogu biti korisni u današnjem svijetu onda moraju biti snabdjeveni sredstvima za prikupljanje informacija iz toga svijeta. Taj zadatak izvršavaju senzori. Djelovanje većine senzora osniva se na pretvaranju nekog fizikalnog efekta u napon koji se nakon toga pomoću analogno-digitalnih konvertora pretvara u binarne podatke (vidi str 12-13)

Najznačajnije tehnike u primjeni senzora, koja se upotrebljavaju u znanosti industriji i vojsci odnose se na mjerenje položaja udaljenosti brzine ubrzanja i sile



Položaj

Često treba odrediti položaj nekog objakta dijela stroja ili u najjednostavnijem slučaju registrirati da li su neka vrata zatvorena ili otvorena što je osnova za rad alarmnih uređaja. To se može izvesti mikrosklopkom, ali kako su takve sklopke podložne greškama i oštećenjima više su u upotrebi beskontaktni uređaji u kojima se prekida zraka svjetla kada je prozor (ili vrata) zatvoren. Još su pouzdaniji magnetski senzori koji rade na principu Hallovog efekta pri kojem magnetsko polje mijenja karakteristiku tranzistora. Da bi se ustanovilo u kakvom je položaju prozor u njega se ugrađuje magnet a detektor koji signalizira stanje kompjutoru u okvir prozora

U mnogim slučajevima kad se želi kontrolirati kretanje nekog dijela stroja najjednostavnije je

pričvrstiti takav dio na klizač promjenljiva otpornika (poput palice za igru — joystick — str 12-13) i mjeriti otpor

Otpornik se međutim može trošiti pa je bolje upotrijebiti optički sustav u kojem se ispred izvora svjetla pokreće mrežica u obliku češlja. Kad se pokretni dio pomakne za širinu jednog zupca češlja kompjutor prima jedan električki impuls. Slično se može mjeriti i rotacija

Daljinski senzori

Udaljenosti i debljine mogu se mjeriti i bez neposrednog dodira s mjerenim objektom. Na str 136-137 opisano je mjerenje čelika u valjaonici, što predstavlja direktnu primjenu ove metode. Jedno od mogućih rješenja sastoji se u emitiranju ultrazvučnih impulsa koji se odbijaju od čelika, a mjeri se kašnjenje impulsa — što je kašnjenje manje, to je veća debljina čelika i obratno. Druga metoda koristi rentgenske zrake koje pri prolasku kroz čelik gube na intenzitetu, a debljina se određuje prema veličini tih gubitaka.

Već se godinama koristi ultrazvuk za određivanje dubine mora mjereći vrijeme koje je po-

magnet
senzor s hal. elekt tranzistorom
prema kompjutoru

Kad kompjutor treba da vodi nadzor nad utroškom energije i nad sigurnošću stena, on mora znati da li su vrata i prozori zatvoreni ili otvoreni. Ako je prozor otvoren, magnet ugrađen u okvir prozora uzrokuje tok struje kroz tranzistor s Hallovim efektom. Tada mali elektronski sklop šalje kompjutoru signal otvoreno-zatvoreno

uteg
svjetlo
mrežica
detektor
prema kompjutoru

Mjerenje položaja poluge u nekoj mehaničkoj napravi, kao što je kontrola položaja zakrilaca u avionu ili noge skakača (vidi str 132—133). Na polugu je pričvršćena mrežica u obliku češlja čiji zubi pomicanjem prekidaju zraku svjetla, a kompjutor prema broju impulsa određuje položaj poluge

I sasvim mali brodovi mogu mjeriti dubinu mora pomoću echo soundera koji šalje zvučni impuls i mjeri vrijeme odjeka od morskog dna, što mnogo olakšava plovidbu



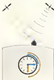
Primjer kako digitalizirani podaci od nevidljivog mogu napraviti vidljivo Razlike u temperaturi dijelova ruke reumatičnog bolesnika prikazane su u različitim bojama Kompjutor pretvara intenzitete infracrvenog zračenja u boje pa se lijepo vide razlike između bolesne (dolje) i zdrave ruke (gore)

trebno da se impuls reflektira od morskoga dna Isti se princip primjenjuje u mjerenju razine kemikalija benzina ili nafte u velikim rezervoarima

Princip mjerenja kašnjenja radio-valova koristi se i u navigaciji Brod ili letjelica može odrediti svoju poziciju mjerenjem razlike u kasnjenju signala što ih prima od nekoliko satelita

Iako kompjutori ne mogu zaista gledati ipak mogu znatno poboljšati ljudsko gledanje Posebni kartografski sateliti snimaju Zemlju s više kamera podešenih na različite frekvencijske pojaseve Slike se analiziraju pomoću kompjutora a iz zbira slika dobije se mnogo više informacija nego iz svake posebno (vidi str 108-109)

Brodovi mogu odrediti svoj položaj na moru pomoću satelitskih signala



Brzine i ubrzanje

Brzinu nije teško mjeriti Brzina vozila mjeri se pomoću senzora koji očitava broj impulsa što ih prima pri svakom okretaju kotača te se iz toga izračunava brzina U avionu se mjeri tlak zračne struje ili radarski signali koji se odbijaju od zemlje a u povratnom signalu Dopplerov pomak uzrokovan kretanjem aviona Za mjerenje brzine strujanja tekućine u cijevi koristi se mali propeler koji na jednom svom kraju ima magnet što pri svakom okretaju pobuđuje Hallov senzor

Brzina se može mjeriti samo u odnosu prema vanjskom svijetu Međutim ima instrumenata potpuno izoliranih od vanjskog svijeta koji mjere ubrzanje i nakon protoka zadana vremena iz te veličine izračunavaju brzinu Mjerači ubrzanja dijele se na dvije grupe jedni mjere linearno a drugi kutno ubrzanje

Za mjerenje linearnog ubrzanja dovoljna je obična vaga Zamislite vagu koju ste s utegom od 5 kg na zdjelici postavili na pod dizala Kad se dizalo pokrene prema gore vaga će u jednom času pokazati teret od recimo 6 kg Na kraju puta kada lift počinje usporavati, kazaljka će pokazivati 4 kg S vagom može biti povezan kompjutor koji će iz dobijenih podataka izračunati sva ubrzanja i brzine
Kutna ubrzanja mjere se žiroskopom Svatko tko se igrao zvrkom zna kako zvrk pruža otpor kad ga se pokuša otkloniti od osi rotacije Taj je otpor mjerljiv i njime se može iskazati kutno ubrzanje Savršeniji žiroskopi koriste laserske impulse koji se šalju po zatvorenoj putanji preko tri ogledala Nakon nekoliko okretaja detektira se impuls ako se vozilo ili neko drugo pokretno sredstvo u koje je ugrađen uređaj okrene dok je zraka bila na putu impuls će stići ranije ili kasnije u detektor a prema tome se onda izračunava stupanj zakretanja Kada se to zna, može se saznati i koliki je bio zaokret, a ako se zna početni smjer lako je odrediti i konačni cilj

Linearni i kutni akcelerometri poput upravo opisanih koriste se za određivanje brzine i položaja interkontinentalnih balističkih projektila Tako su precizni da nakon leta od nekoliko tisuća kilometara bojeva glava pogađa cilj s mogućom greškom ne većom od nekoliko stotina metara

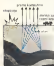
Sila

Najlakše je izmjeriti silu ako se stlači opruga i izmjeri njena dužina Ali zamislite da treba izmjeriti silu potiska motora mlaznoga aviona Međutim i to se može izvesti na sličan način jedino što će ulogu opruge preuzeti potporanj koji će sila mlaznog motora tlačiti kao i oprugu ali u mikroskopskim razmjerima Da bi se ovo elektronički koristilo treba na potporanj prilijepiti komadić metalne folije koja će djelovati poput otpornika što je pritisak veći to će se traka više širiti i njezin će otpor porasti, a kompjutor će iz tih vrijednosti lako izračunati potisak motora

Barem polovica tajnovitosti kojom je bila obavijena plovidba dalekim morima odnosila se na kompliciranu navigacijsku aritmetiku Ali kompjutor je uspio uništiti i ovu tajnu

Kompjutor na brodu obavlja mnoge korisne poslove ali pod uvjetom da je opskrbljen brojnim ulaznim podacima Osnovne veličine koje treba izmjeriti jesu brzine broda smjer plovidbe realtivna brzina i smjer vjetra Prikupivši podatke pomoću instrumenata prikazanih na slikama desno kompjutor će obaviti mnoštvo korisnih poslova Prva briga navigatora je određivanje položaja broda i ako je vidljivost dobra on će se osloniti na svjetionike i konfiguraciju obale Ali kada se ništa ne vidi na horizontu po noći ili magli tada treba približno odrediti položaj broda to jest prikaz kursa i brzine od polazišta do trenutne pozicije što će kompjutor op-

Dolje, geologe zanima raspored kamenih slojeva ispod zemlje To se može izvesti s više mikrofona koji snimaju odjek eksplozije što se odbije od različitih geoloških slojeva Kompjutor će kasnije prema snimkama odjeka rekonstruirati sliku rasporeda podzemnih slojeva



U samoposluživanju



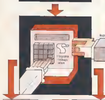


čitač prugastog kôda

U kompjuteriziranom samoposluživanju sve počinje s prugastim kôdom (bar code) otisnutim na svakom artiklu. Čitač (u kojem se zraka svjetla reflektira od otisnutih pruga i pada na senzor) daje kompjutoru podatke o cijeni artikla i njegov opis što blagajna automatski otiskuje na računu. Kupac plaća na uobičajeni način ali kompjutor odmah knjiži uplatu i ažurira zalihu. Ako na zalih više nema tog artikla kompjutor će naručiti novu količinu mnogo brže nego što bi to uradio čovjek.

U daljem razvoju predviđa se da će kupac svoju kreditnu karticu uložiti u blagajnu a ona će preko kompjutora poslati narudžbu banci da iznos njegova računa prebaci na račun firme. Prodavaonice i banke zajednički rade na tome da eliminiraju gotov novac iz opticaja jer je skup i lako se gubi.







Jutro na farmi



Kompjuterizirane mljekarske farme počinju pokazivati svoju vrijednost. Svaka krava nosi oko vrata mali pasivni radio odašiljač pa kada uđe u muzilište kompjutor je identificira i registrira količinu hrane. Kompjutor može kontrolirati zalihe i prodaju mlijeka. Ako ima podatke o livadama na kojima krave pasu on može pomoći da se odredi najpovoljnija količina inače skupih fertilizatora.











Revolucija koja je zahvatila zrakoplovstvo ne vidi se izvana na avionima, ali su promjene tim veće iznutra, u pilotskim kabinama.
Pilotska kabina u britanskom avionu Aerospace 1-11. Na lijevoj strani vide se novi displeji umjesto tradicionalnih brojčanika s kazaljkama. Kompjutor upravlja dvama velikim monitorima u boji: na lijevom je vidljiv umjetni horizont, a uz njega visinomjer i pokazivač stupnja penjanja.
U vojnim avionima se slika takvog displeja projicira na staklo vjetrobrane, pa izgleda kao da se nalazi u zraku, u vidnom polju pilota. Sa HUD (Head-up Display) displejom mogu se prikazati radarske slike neprijateljskih aviona koji se ne vide prostim okom ili skrivenih i kamufliranih ciljeva na zemlji. Isti se princip primjenjuje i u civilnom zrakoplovstvu gdje se u vidno polje pilota projicira trodimenzionalna radarska slika aerodromske sletno poletne staze, tako da može slijetati i polijetati bez obzira na vidljivost.

Putnički avion je neizmjerno komplicirana množina raznih naprava koje se moraju pažljivo posluživati da bi zrakoplov sigurno stigao na pravo mjesto i u pravo vrijeme uz minimalni potrošak goriva. Na početku leta posade opskrbljuje kompjutor podacima plana leta, a nakon polijetanja kompjutor preuzima nadzor nad daljim penjanjem, optimalnom visinom i brzinom zrakoplova. Ako treba povećati visinu, kompjutor će odabrati pravi čas za taj manevar, a pri dolasku će odrediti najbolju točku za početak spuštanja.

skrbljen potrebnim podacima izračunati u tili čas

Navigatorima je dragocjeno znati apsolutnu brzinu i smjer vjetra što kompjutoru uopće nije teško izračunati ako je poznata brzina plovidbe smjer prema kompasu, prividna brzina i pravac vjetra

Murphyjev zakon primijenjen na jedrenja morem, tvrdi da se tri četvrtine vremena troši na mučna i polagana savladavanja privjetrine Najviše dakako ovisi o kormilarevoj spretnosti da zna najbolje iskoristiti vjetar ali nije lako ustanoviti koliko je zaista uspio I ovdje može pomoći kompjutor koji će izračunati brzinu broda uz vjetar i iskazati rezultat u postocima Sve mogućnosti uporabe kompjutora time još uvijek nisu iscrpljene Kada obala nije na vidiku navigatori moraju određivati poziciju broda prema suncu i zvijezdama Najprije treba sekstantom izmjeriti potrebne kutove, a onda pomoću nespretnih tablica načiniti vrlo komplicirane proračune da bi se znao položaj broda Kompjutor će umjesto njih sve to obaviti puno preciznije, i dakako neusporedivo brže





Lijevo mali senzor montiran na vrhu jarbola dojavljuje brodskom kompjutoru brzinu i pravac vjetra

Gore brodski kompjutor mjeri relativnu brzinu broda pomoću malog propelera pričvršćenog ispod brodske oplate



Ako čovjek padne u more malo je nade da ga se spasi jer je teško točno odrediti put kojim je prošao brod Ali ako brod ima kompjutorom upravljeni navigacijski uređaj na povik „čovjek u moru!“ treba samo pritisnuti tipku kojom se uređaj resetira (postavlja na nulu) okrenuti brod i kormilariti do mjesta na kojem će displej pokazivati opet nulu Na tom mjestu mora da se nalazi čovjek u moru

SERVO PETLJA

Primanje i obrade podataka iz vanjskog svijeta je vrlo važan segment u kompleksu kompjutorskog upravljanja strojevima ali ne i najvazniji Glavnu ulogu u tome ime servo petlja Da bismo razumjeli taj pojam razmotrimo jednostavni primjer: pretpostavimo da vozite auto na potpuno ravnoj cesti

Mnogo štošta će vas ometati da vozite po ravnoj liniji upravljački mehanizam može zanositi na jednu ili drugu stranu, možda na cesti ima neravnina koje skreću vozilo zatim naleti vjetra i struja zraka od velikih kamiona u prolazu — sve to može skretati vozilo s njegove putanje

Posve je jasno da se ne može napisati program za upravljanje automobilom jer se unaprijed ne može točno znati ponašanje ostalih sudionika u prometu Ako neki kamion prođe pored vašega auta sekundu ranije ili kasnije program može biti smrtno opasan Ono što se traži je nešto mnogo fleksibilnije nešto što se može prilagoditi promjenljivim i nepredvidivim okolnostima

Suočen s ovakvim problemom kompjutor se mora ponašati poput čovjeka tokom vožnje mora eksperimentirati Kad vozite nepoznati auto Iskušavate volan i kočnice da biste ustanovili kako se ponašaju Kompjutor mora raditi to isto ako se vjetar promijeni mora osjetiti promjenu i prilagoditi pritisak na volan Tehnika kojom se kompjutor služi u takvim situacijama zove se servo petlja (servo loop)

Kad vozite po ravnoj cesti mislite da ništa ne radite ali to je samo privid Naprotiv vi neprestano djelujete prema pravilima servo petlje Evo kako stalno promatrate cestu i smjer vožnje određujete prema znakovima kao što su rub i središnja linija ceste te kretanje ostalih vozila Izračunavate razliku između željenog i stvarnog položaja vozila i zakrećete volan toliko da se ta razlika svede na nulu Drugim riječima ako je vozilo skrenulo malo u lijevo vi ćete volan okrenuti malo u desno Ako ste pak jako skrenuli u lijevo ili predstoji desni zavoj volan okrećete jako na desno Program napisan na Microsoft BASICu simulira ove postupke

```
  5 K1 - 05 K2 -   2 K3 - 8 K4   210 INPUT
    WIND MPH  W
 20 PRINT  WHEEL ANGLE  DEGREES
 40 PRINT TAB(25)  CAR DEVIATION  FEET
 50 FOR J - 1 TO 20
 60 PRINT USING  ***  *  A*10
 70 PRINT TAB(25) PRINT USING  ***  *  I
 80 D=K1*W  A
 90 I - I+D
100 A1  K2*(I + K3*D)
110 A - A+K4*(A1  A)
120 NEXT J
130 GOTO 10
140 END
```

U programu su određene četiri konstante K1 se odnosi na smjer vožnje i utjecaj vjetra K2 i K3 na vozačevo opažanje odstupanja od željenog smjera a K4 na vrijeme reagiranja Nakon toga program traži upis brzine vjetra a ovisno o smjeru iz kojega puše predznak će biti + ili — Slijedi ispisivanje tablice iz koje se vidi kut kotača u odnosu prema sredini ceste i udaljenosti vozila od zadanog smjera U proračunu dakako nisu predviđeni svi fizikalni faktori

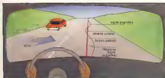
Svaki put kad se J u petlji poveća to odgovara vremenskom intervalu od recimo 1/10 sekunde Linija 60 ispisuje kut kotača pomnožen s 10 da bi se dobila zamjetljiva veličina a linija 70 ispisuje odstupanje od željenog smjera Linija 80 izračunava D dodatno odstupanje koje se pojavilo nakon zadnjeg okretaja petlje D je proporcionalan konstanti K1 pomnožen s brzinom vjetra W zbrojen s kutom kotača A (koji obično ima suprotan predznak)

Linija 90 daje novo odstupanje prethodnom zbroju I Program pretpostavlja da vozač čim primijeti da se nešto novo događa odmah čini što je potrebno No zbog tromosti našeg oka, mozga i mišića A se ne može odmah prilagoditi novim uvjetima pa program to simulira izračunavanjem međukuta A1 u liniji 100 Proučavanja ljudskih reakcija u sličnim situacijama pokazuju da je vrijeme reagiranja proporcionalno (preko konstante K2) odstupanju I nešto povećano ovisno o stupnju promjene I To je zapravo D ili vrijednost za koju se I promijenio nakon posljednjeg okretaja petlje pomnožen s konstantom K3

Linija 110 tada mijenja A za iznos proporcionalan razlici između A i A1 čime se uračunava kašnjenje vozačeve reakcije pomoću konstante K4 Na slici desno prikazana je tablica u početnim uvjetima vozilo na pravcu i udari vjetra od 30 milja na sat što otklanja auto za 1 9 stopa (60cm) niz vjetar Međutim u idućem okretaju petlje kotači se zakreću za 19 9 stupnjeva na suprotnu stranu te dovode vozilo na udaljenost od pola stope od zadanoga smjera Vozilo zatim ponovo skreće ali se i smiruje na udaljenosti od 0 8 stopa od zadane linije

To je ponašanje tipično za mnoge servo uređaje U početku je odstupanje veće ali se tokom vremena sve više smanjuje i stabilizira na nekoj maloj vrijednosti Ako se na primjer konstante K4 (konstanta vozačeve reakcije) postavi na 0 4 (standardna brzina reakcije vrhunskih sportskih vozača) skretanje se skoro trenutno korigira Ako se poveća na 0 5 vozilo se uopće ne smiruje nego stalno vrluda s jedne strane ceste na drugu a pogotovo ako se poveća na 0 6 Tada sve brže šeće od jednog do drugog ruba ceste dok se ne prevrne ili ne nađe pod kotačima kamiona koji dolazi iz suprotnog smjera

Služeći se servo petljom kompjutor može ignorirati mnoštvo činjenica iz vanjskoga svijeta Zapravo on pomalo eksperimentira kad treba izdati neku naredbu pa je zato potrebno da unaprijed zna koliku će snagu koristiti za uprav-



NAGIB KOTAČA I OTKLON VOZILA	ODSTUPANJE U STOPAMA (30 5cm)
0 0	0 0
—11 3	1 5
—18 0	1 9
—19 8	1 6
—18 6	1 1
—16 5	0 7
—15 0	0 6
—14 3	0 6
—14 3	0 7
—14 6	0 7
—14 9	0 8
—15 1	0 8
—15 1	0 8
—15 1	0 8
—15 0	0 7
—15 0	0 7
—15 0	0 7
—15 0	0 7
—15 0	0 7
—15 0	0 8

ljanje jer bi inače mogao tako jako zakrenuti kotače da bi se vozilo sigurno prevrnulo

Djelovanje servo petlje ovisno je o negativnoj povratnoj sprezi tj o pojačavanju signala razlike između onoga što se događa i onoga što bi se trebalo dogoditi Princip servo petlje automatski uzima u obzir sve osobine vozila i ceste, sve neravnine puta i nesavršenosti vozila Kad ne bi bilo tako, kompjutor bi bio sasvim nepodesan za upravljanje bilo kojim strojem

Programi niže razine kao što je ovaj, mogli bi se kontrolirati pomoću programa viših razina Na primjer najveća moguća brzina vozila određivala bi se prema optičkoj vidljivosti Ako je pred vama ravni komad ceste koja se proteže sve do horizonta onda se može voziti brzinom od 200km na sat Ali u času kad pred vozilom iskrsne zavoj brzinu treba smanjiti na 40km na sat Dakako na cesti ima i drugih vozila pa i njihove brzine i udaljenosti treba uzeti u račun pri određivanju vlastite maksimalne brzine

Kompjutor sa servo petljom možemo zamisliti kao jednostavnu crnu kutiju sa dva ulaza — za upravljački signal i signal odstupanja — i jednim izlazom te nekoliko tipaka za unošenje konstanti Kutija ili bolje reći potprogram može sa svime upravljati U ovom slučaju radi se o upravljanju automobilskim upravljačem ali on može također isto dobro upravljati i papučicom za gas uzimajući u obzir i obrađujući različite konstante



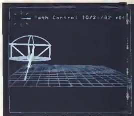
SKAKAČ

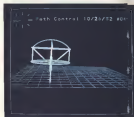
Razvoj jeftinih ı mocnih mikrokompjutora omogucuje ızradu takvih strojeva o kakvima se nekad moglo samo maštati Za primjer neka posluži jednonogi skakač vrst mehaničkog klokana sagraden za proučavanje kretanja ljudi i životinja ali ı za ıstraživanje mogućnosti proizvodnje vozila koje se može kretati bez kotača ili gusjenica

Pomocu mikroprocesora konstruiran je uređaj koji može stajati usprevno bez obzira na vlastitu geometriju Sastoji se od 'tijela' — okvira u koji su smješteni različiti osjetljivi instrumenti naprave ı kompjutori — I 'noge' vrlo pokretljive pneumatske naprave s malim koeficijentom trenja ı stopalom koje ne klizi

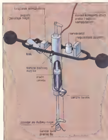
Buduci da bi vlastiti izvor energije bio pretežak skakač prima stlačeno ulje ı zrak preko lagane gumene cijevi Noga mu je kardenskim ovjesom pričvršcena na sredini tijela ı može se kretati u svim pravcima — napred natrag lijevo I desno e posebni senzori šalju kompjutoru podatke o točnom položaju noge u kojoj se nalazi pogonska sila cijele naprave Kada se noga naglo produlji stroj skače uvis a u doskoku se noga opet skraćuje Treba vrlo malo zraka da bi se nadoknadio gubitak energije zbog trenja mehaničkih dijelova te stroj može skakati na jednom mjestu u ritmu vlastite rezonentne frekvencije

Ako stroj može skakati u mjestu a da ne padne — razmišljali su konstruktori — neće biti teško natjerati ga da se krece Ali da bi skakutao poput pravog klokana, upravljački program mora konstiti tri servo petlje (vidi str 130-131) Prva petlja kontrolira visinu skoka ispuštanjem ili dodavanjem zraka preko ventila pneumatike Težine skakače ı snage pneumatskog uređaja





Jedna od prvih verzija skakača koji treba bočni potporanj da bi mogao funkcionirati

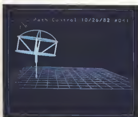


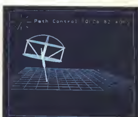
tako su usklađene da tvore rezonentnu cjelinu što znači da se veci dio energije za skok dobija iz prethodnog skoka Kompjutor uspoređuje podatke koje dobija od senzora visine s unaprijed zadanom idealnom visinom skoka pa ako je potrebno naređuje zračnom ventilu da uzme ili ispusti zrak iz cilindra

Druga petlja izračunava kut potreban nozi za siguran doskok uzimajuci u obzir brzinu kretanja ı nagib tijela Proizlazi da ova petlja upravlja svim funkcijama kretanja bilo da stroj poskakuje u mjestu trči ili se kreće jednoličnom brzinom bilo da preskače zapreke ili usporava kretanje Osim toga prima žiroskopske signale u kutnom ubrzanju ı relativnom položaju noge ı tijela te pri skakutanju u mjesta ispravlja sitne nestabilnosti ravnoteže Zadatak trece petlje je da stabilizira napravu dok miruje

Kada sve tri petlje ispravno funkcioniraju skakač može vrlo dobro imitirati klokanov skokoviti trk Konstruktori namjeravaju izraditi na-

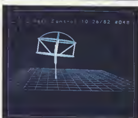












pravu koja će funkcionirati po istom principu ali sa četiri odnosno šest nogu namijenjenu za kretanje po neravnim terenima gdje su vozila s kotačima ili ona koja lebde na zračnom jastuku nemoćna Kao što smo vidjeli na str 116-119 kompjutorski vid ipak još nije dovoljno razvijen da bi mehaničke životinje imale upotrebljive oči pa će i dalje čovjek određivati kamo da stave koju nogu

Na ovih šest fotografija prikazana je kompjutorska simulacija usavršene verzije skakača (vidi tekst) pomoću koje su provjereni konstrukcijski principi i koja je omogućavala brzo i jednostavno unošenje promjena. Izgradnja stroja bila je potrebna samo da bi se utvrdilo nije li u programu simulacije bilo nekih grešaka



Pet faza skakačeva skoka superponiranih preko slike klokana

Umjetnička vizija laboratorija za gradnju androida. U stvarnosti su se svi pokušaji izgradnje mehaničke simulacije ljudskoga tijela pokazali daleko težima nego što je itko mogao pretpostaviti.
Na fotografiji dolje desno je pokojni Chris Evans s robotom Freddyjem snimljen na edinburškom Sveučilištu. Iako primitivan, Freddy je pokazivao mnoga svojstva koja očekujemo od robota budućnosti. Slika je snimljena 1975. a neposredno nakon toga vlada je odlučila da prekine robotička istraživanja u korist nuklearne fizike. Rezultat je te kratkovidne odluke da Britanija danas kupuje Freddy jeve potomke od Japanaca.



Malo tko se osobno susreo sa samostalnim ilmenim robotom poput onoga R2D2 iz Rata zvijezda', ali u mašti ih susrećemo u gotovo svakom djelu znanstvene fantastike. Ima ih svakojakih pedantnih i nepogrešivih patetičnih i pokvarenih robota-špijuna i robota-luđaka koje bi trebalo poslati u topionicu jer su iznevjerili Asimovljeva tri zakona robotike.

Kao što ćemo vidjeti konstruiranje klasičnih ilmenih robota još uvijek je nedokučivo i takvo će ostati vjerojatno još nekoliko desetljeća. Zasad nitko ne zna kako bi trebalo pristupiti gradnji čovjekolikih robota androida poput onih iz znanstveno-fantastičnih filmova. Roboti što danas postoje tek su dio složenih industrijskih postrojenja.

Kada bude postojao kompjutor koji će svojim senzorima kontrolirati vanjski svijet, roboti će na mnogim poslovima zamijeniti ljude. Nije to ništa novo. Na početku industrijske ere parni stroj je bio još tako nesavršen da je trebalo ručno upravljati ventilom za dovod i ispuštanje pare: kada se klip našao u gornjoj mrtvoj točki trebalo je paru iz kotla pustiti u cilinder, a kada bi došao u donju mrtvu točku trebalo je paru ispustiti. Tvorci stroja smatrali su da uz stroj treba unajmiti i jednog dječaka koji će u pravi čas otvarati i zatvarati ventile.

Priča kaže da se neki momak dosjetio jedu i ventil privezao uzicom za radilicu te tako riješio svoj problem popodnevnih sastanaka s djevojkom. Nije ni slutio da je napravio prvog robota u

kojem se očituju sva tri osnovna principa robotike što vrijede i danas: opažanje, inteligencija i izvršavanje.

Zadaća robota je da odgoneta podatke koje prima iz vanjskoga svijeta i odlučuje što će s njima uraditi. U slučaju parnog stroja trebalo je u pravi trenutak otvoriti ili zatvoriti pravi ventil i upotrijebiti za to primjerenu snagu. Proces opažanje — izvršavanje bio je spojen uzicom. U većini ostalih slučajeva opažanje i izvršavanje su dvije potpuno odvojene funkcije — a između njih danas stoji kompjutor.

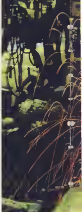


ROBOTI U INDUSTRIJI

Na str. 124-129 vidjeli smo kako kompjutor prima podatke iz vanjskoga svijeta pomoću mnoštva raznih senzora, od jednostavne sklopke na prozorskom okviru pa do cijelog lanca radara. Čim dobije ulazne podatke kompjutor izvršava ono što se od njega očekuje, ali kako još uvijek nismo u stanju napraviti kompjutor koji vidi i čuje poput nas, za sada se ključni problem robotike svodi na dobavljanje ulaznih podataka koje ća kompjutor razumjeti.

Ako želimo da robot upravlja valjaonicom čelika, nije dovoljno usmjeriti televizijsku kameru na valjke po kojima klizi uzareni čelik i očekivati da će kompjutor samo iz slike dokučiti debljinu čelične ploče. Treba montirati još i posebne uređaje koji će s dovoljnom točnošću mjeriti debljinu čelika i slati je u pogodnom obliku kompjutoru.

Desno: fotografija robota-zavarivača snimljena u Chryslerovim pogonima rječito dokazuje da takva mjesta nisu dostojna čovjeka.



Dolje: u jednoj japanskoj tvornici robota zabrinuti robot niskog kvocijenta inteligencije osamljen luta naokolo u potrazi za poslom.

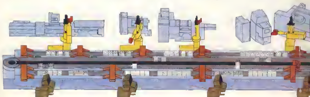


Kada izmjeri debljinu čelika, kompjutor odlučuje na koju će mjeru podesiti razmak valjaka golema preše i upravljati servo uređajem. Ja lako zamijeniti iskusnog radnika koji je do sada ručno obavljao taj posao. Cijeli sistem, mada ne odgovara našoj uobičajenoj slici robota, zapravo je pravi robot, poput mnogih drugih kojima je napučena moderna industrija.

Mnogi od njih su usko specijalizirani i njihove adaptabilne mogućnosti su vrlo male. Na primjer, robot koji upravlja valjaoničkom prešom može proizvoditi samo čelične ploče razne debljine. Ako bismo htjeli da proizvodi željeznička tračnica, bio bi potreban cijeli tim stručnjaka da ga preprogramira. Robot nema one fleksibilnosti što je posjeduje čovjek kojega je zamijenio. Ali s druge strane, on ne griješi, ne uzrujava se, nikada nije umoran i zlovoljan, a i troškovi za njegovo održavanje su manji.

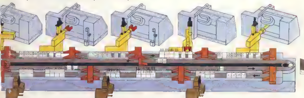
Konstruktori robota se trude da proizvedu stroj koji će biti toliko specijaliziran da savršeno obavlja svoj posao, a ipak dovoljno fleksibilan da bi se mogao preprogramirati bez mijenjanja strojne opreme. Taj je problem posebno naglašen kod robota koji rade na traci i od kojih se očekuje da vide dijelove što dolaze. Ti su roboti

Dolje: automatizirana proizvodna linija, na kojoj se od neobrađenih komada metala izrađuju zupčanici i osovine što ih pokretna traka donosi do određenog stroja. Svaki stroj poslužuje po jedan robot koji uzima komade metala sa trake, ulaže ih u stroj i kad budu obrađeni, vraća ih na pokretnu traku. U prednjem planu vide se kontrolne postaje za svaku radnu operaciju. Kompjutor koji upravlja proizvodnom linijom nalazi se u pokrajnjoj prostoriji.



opremljeni posebnim senzorima u kojima se prekida zraka svjetla kad na nju naiđe novi dio. Kad se pređe na dijelove drugačijeg oblika senzori sa moraju iznova podešavati a cijeli stroj preprogramirati. Drugi je problem što još uvijek nismo kadri napraviti tako osjetljivu i preciznu napravu kao što je ljudska ruka. Ruka robota može se konstruirati da radi samo sa štrcaljkom za boju, s aparatom za zavarivanje ili sa bušilicom, ali ona zna upotrebljavati samo onaj alat za koji je konstruirana.

Bilo kako bilo, roboti su počeli zamjenjivati ljude na teškim i monotonim poslovima, u opasnim i nezdravim uvjetima. Izvrsno mjesto za robota je lakirerski i zavarivački posao u automobilskoj industriji i već je postalo sasvim normalno da ih ondje nalazimo. Počeli su se već spuštati i u rudnike da bi oslobodili čovjeka od rada neadekvatnog njegovoj konstituciji.



KAKO RADI ROBOT

Svaki sposoban radnik može sastaviti vrlo komplicirani stroj ako ima nacrt i sve potrebne dijelove. Zamislite s kakvim se poteškoćama susreća konstruktor kad konstruira robota koji bi trebao obavljati isti posao. Treba za to u prvom redu imati neki uređaj za gledanje koji bi mogao identificirati svaki dio, zatim inteligentni ekspertni sustav koji bi znao čitati nacrt i konačno ruke koje se mogu služiti raznim alatima toliko osjetljive da mogu sastaviti sat i dovoljno snažne da lome stijene.

Cijela bi stvar morala biti jeftinija i jednostavnija za programiranje od živoga radnika. Može se po prilici izračunati da je cijena prosječnog robota jednaka petogodišnjoj prosječnoj plaći kvalificiranog radnika na Zapadu. Dakako, dobar robot mogao bi koštati i više, jer on može više proizvoditi ili pak raditi na opasnijim i težim poslovima nego čovjek, ali današnja tehnologija još uvijak nije u stanju napraviti čovjakolikog robota koji bi mogao potpuno zamijeniti sposobnog radnika. Ono što zasada nazivamo robotom, tek je jedna vanjska jedinica kompjutora koja može samo dodavati stvari ili obavljati samo jednu jedinu radnu operaciju na proizvodnoj traci.

Glavni dio prosječnog robota je njegova 'ruka' koja se krece u svim smjerovima u radijusu oko 3 metra. Na kraju ruke je zglob koji se može vrtjeti i zakretati na sve strane. Umjesto univerzalne ruke, većina robota ima ugrađena posebna alate, kao što je aparat za zavarivanje ili recimo, štrcaljka za boju.

Zadaća konstruktora je da upravljački kompjutor snabdije jezikom koji će krajnjem korisniku omogućiti jednostavno programiranje robota. Korisnik će, recimo, narediti robotu da čeka dok se ispred njega ne nađe automobilska karoserija što putuje na pomičnoj traci. Kada karoserija prekine zraku svjetla, robot će to shvatiti kao znak za početak varenja. To uključuje i naredbe 'Pomakni se iz početnog položaja na početak šava i vari', ili 'Kraj šava. Vrati se na početni položaj'.

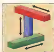
Naredba 'vari' je zapravo potprogram koji raspoznaje udaljenost elektroda za varenje od karoserija, kontrolira tok struje, dužinu elektrode i

Najvažnije osobine industrijskog robota je pokretljivost ruke i alata koji drži u njoj u svim smjerovima i u polukrugu oko svoga postolja. Budući da se ruka mora pokretati brzo i precizno, a pri tom podizati i teške terete, cijela konstrukcija mora biti vrlo čvrsta i stabilna. Normalni robot ima šest stupnjeva slobode (prikazani su strelicama), što znači da se kretanje ostvaruje u šest zglobova. To su struk, rame, lakat, dva zgloba u zapešću i držač alata koji se može okretati oko vlastite osi. Na slici su prikazana četiri alata (po stoje i mnogi drugi) s kojima se roboti služe. To su glava za električno zavarivanje, glava za točkasto zavarivanje, mala hvataljka i vakuumska hvataljka za veće plosnate dijelove. Na lijevoj strani vidi se upravljački kompjutor.





Postoje četiri osnovna načina da se pokreću zglobovi robotove ruke. Stroj prikazan na velikoj slici služi se načinom prikazanim na četvrtoj slici.






sve ostalo što spada u postupak zavarivanja. Naredbe „početak šava" i „kraj šava" odnose se na položaj ruke u prostoru, što se unosi u program prema obliku i veličini karoserije, a programer može podatke unijeti u program kao koordinate ili postupkom učenja, voditi ručno ruku robota po zadanom putu. Jezik mora također sadržavati potprogram za izglađivanje krivudanja do kojih dolazi pri ručnom vođenju robotove ruke.

Upravljački kompjutor također mora biti opskrbljen potprogramima koji će robotu omogućiti da neke jednostavnije radne operacije ponavlja na različitim mjestima. Pretpostavimo da je robot programiran za stavljanje boca u sanduke i navrtanje čepova. Pokret navrtanja čepa uvijek je isti, ali se svaki put odvija na drugom mjestu.

Kada senzori postanu još savršeniji, roboti će se moći programirati sasvim općenitim programskim jezikom. Izdavat će im se iste naredbe poput onih što se daju sirotom ljudskom biću: Skini te proklete boce sa traka i navrni čepove na njih, glupane!

ROBOTI ZA VJEŽBU

Vrlo zanimljiva, ako ne i egzotična vanjska jedinica za mikroračunalo je robot za vježbu, mala stolna verzija monstruma iz tvorničkih hala. Jedan od ciljeva mikrokompjutorske industrije je širenje znanja o kompjutorima i njihovoj primjeni u svim oblicima i prilikama, pa ipak ovaj mali robot nije tek puka tehnička egzibicija, već vrlo korisna naprava, pogotovo kada treba rukovati s mnogo malih predmeta, a u nekom manjem poduzeću mogla bi se s njim organizirati prava proizvodna traka.

Stroj je proizvela, i to vrlo solidno, tokijska tvrtka Mitsubishi. Opremljen je vlastitim kompjutorom sa standardnim CP/M operacijskim sustavom, koji se može povezati s bilo kojim kompjutorom preko serijske porta.

Ruku robota pokreće šest malih stepenastih motora, poput onih koji pokreću i velike industrijske robote. To su motori koji reagiraju na serije električkih impulsa. Svaki impuls uzrokuje određeni zakret osovine motora, koji se preko prenosnog mehanizma pretvara u pokret robotove ruke, što iznosi između 0.04 i 0.08 stupnjeva po impulsu. Upravljački kompjutor određuje položaje svakog članka prema broju poslanih impulsa. Na taj način robot ne mora imati niz kompliciranih i skupih senzora koji bi mjerili i dojavljivali položaje njegovih zglobova.

Motor smješten u kućište okreće robota oko njegove vertikalne osi. Zglobovi ramena, lakta i zapešća zakreću se horizontalno, tako da hvataljka doseže do svake točke u prostoru oko robota, a zglob zapešća se, za razliku od ljudskoga, vrti u punom krugu oko svoje osi. Šesti motor namatanjem čelične uzice zatvara krakove hvataljke.



Naredbe koje upravljaju robotom slične su onime za pnnter ı izdaju se u obliku stringova Na primjer ako je robot priključen na kompjutor preko porte za pnnter naredba na Microsoft BASICu koja glası

```
LPRINT"H'
```

znači da treba svih šest motora postaviti u početni položaj (H = HOME — kući naredba kojom se prije neke operacije određuje početni položaj Op prev) Upravljački kompjutor tako zna točan položaj odakle je ruka kranula I također kontrolira položaj ruke u svakom trenutku operacije

Kompjutor u podnožju robota uzima u obzir vrijeme potrebno za pokretanje ı usporavanje svakog zgloba pa su svi pokreti glatki I precizni

Stroj razumije 15 naredbı podijeljanih u dvije razine Na nižoj razini svaki se zglob okrene za određeni broj koraka Naredbe se mogu unositi preko tastature ili pomoću tastera na samom robotu koji ga uči pokratima što ih treba izvoditi Na pnmjer na početku programa za igru križić-kružić, koja se vidi na slici, 'ruku' treba dovasti do sredine ploče pokazati joj gdje se nalaze figunce ze igru ı predvidjeti sve pokrete tako da slijepi stroj može obaviti sve što je potrebno tokom igre

Kada se utvrdi položaj ruke u prostoru koordinate se unose u mamoriju kompjutora koji ih pamtı ı dovodi ruku na taj polozaj kad god primı naredbu

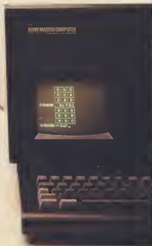
```
LPRINT 'M3
```

Kada dođe u zadani položaj hvataljka se zatvara podiže figuricu odnosi je na drugi položaj ı ispušta Ne bı bilo teško napisati kakav koristan program na primjer za hvatanje malih predmeta recimo epruveta ı slagati ih u kutije

Upravljački program treba dakako pnpremiti robota za posao Treba sastaviti listu položaja ruke u obliku određenog broja koraka za svaki motor posebno ı sve to unijeti u mamoriju Na primjer program križić-kružić počinje naredbama za zauzimanje niza polozaja

```
10 LPRINT   P1.0,372 —958.592,—592 0"
20 LPRINT   P5 739 —707 —431 —86
—1114 0"
```

ı tako dalje Hvataljka se pomiču na određeni položaj u toku programa jadnostavnim navođenjem odgovarajucih brojaka



ANDROIDI

Limeni ljepotani (i ljepotice) imaju etabliranu ulogu u našoj fantaziji. Svakome bi se svidio prijateljski pouzdani domaći robot koji će kad zatreba donijeti i kavu (dolje). Stvarnost je međutim manje pitoma kao što se vidi na prototipu robota Shakeyja (u sredini) koji je sagrađen na Sveučilištu Stanford

Na suprotnoj strani
Gore desno Andy Warhol je u stvari robot vrijedan 40000 dolara, načinjen za jedan TV šou

Dolje desno Čudesna igračka — robot Robi (1971) Eduarda Paolozzija. Čini se kao da se život pretočio iz ovog supersoničnog djeteta u njegova uglatog prijatelja

Gore lijevo inspiracija Fritza Langa u filmu *Metropolis* (1926) bio je strah od pretjerane mehanizacije suvremenih gradova i ova čvrsta ali ne baš previše ugodna dama

Dolje u sredini Ideja da se roboti prikažu kao ljudska bića ali neusporedivo snažnija brža i opasnija našla je odjeka i u filmu *Na oštrici noža* (Blade runner)

U neku ruku svaki je kompjutor vrst androida, jer oponaša ljudske mentalne procese. Ali kompjutor ne izgleda kao čovjek nedostaje mu sposobnost kretanja ne može baratati predmetima nema vida sluha ni opipa. Mnogi kompjutorski znanstvenici baš kao i umjetnici trudili su se da stvore stroj koji bi imao ako ne sve a ono barem dio ljudskih osobina. U toj težnji najdalje su dotjerali proizvođači filmova koji su najčešće pošli najjednostavnijim putem koristili su ljude da glume čovjekolike robote. Iako je na ovom polju uloženo mnogo napora na kraju su mnogi ipak priznali da je naše znanje u usporedbi s majkom Prirodom vrlo skromno

Android bi trebao biti stroj sposoban da se kreće po neravnu tlu da se penje uza stepenice po drvaću i stijenama morao bi biti kadar podići i nositi teret po prilici vlastite težine a istim bi rukama morao navesti konac u iglu. Današnja tehnologija ne raspolaže sredstvima koja bi makar izdaleka bila nalik ovim osobinama. Kada bismo znali (a još uvijek ne znamo) napraviti nogu koja bi mogla hodati težina bi joj se prije mogla mjeriti tonama nego li kilogramima a baterije koje bi stavljale u pogon ovakav stroj brzo bi se iscrpile

Sposobnost razlučivanja ljudskog oka mogle bi se usporediti s televizijskom slikom od 3 milijuna piksela a najbolje televizijske kamere raspolažu danas s jednim milijunom. Pa čak kada bismo i imali optičku napravu ravnu ljudskom oku obrada primljenih informacija trajala bi satima

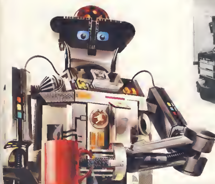
Ljudski mozak sadrži 10 000 milijuna međusobno povezanih neurona. Ne znamo koliki je kapacitet jednog neurona u kompjutorskim omjerima, ali se može pretpostaviti da odgovara memoriji od po prilici 100 bajtova što znači da bi mozak imao kapacitet od milijun milijuna bajtova. To bi odgovaralo kocki ispunjenoj suvremenim memorijskim čipovima čije bi stranice bile dugačke 30 metara. Kad bi i postojala takva memorija trebalo bi je znati organizirati jer bi suvremenom 16-bitnom mikroprocesoru trebalo tri tjedna da u ovako velikoj memoriji pronađe riječ od četiri slova

To su razlozi zbog kojih se treba pomiriti s činjenicom da su androidi stvar daleke budućnosti. Zanimljivo je međutim da je prije deset i više godina bilo ozbiljnih pokušaja na tom polju. Konstruktori su se služili za ono vrijeme vrlo moćnim kompjutorima ali je teškoća bila u tadašnjem nedovoljno razvijenom softveru. Pokusi su dokazali da između mogućnosti stroja i čovjeka stoji preduboki ponor preko kojega ne bi imalo smisla graditi most barem ne na ovom stupnju razvoja

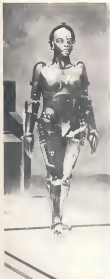
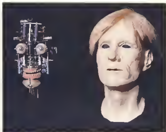
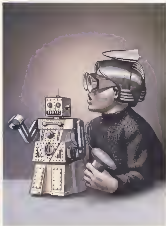
Tada su bila sagrađena dva eksperimentalna androida. Jedan na Sveučilištu Stanford nazvan Shakey, a drugi sagrađen na edinburškom Sveučilištu zvao se Freddy. Shakey (Klimavko) je bio jednoruki pokretni robot s ugrađenom televizijskom kamerom koji se klimavo kotrljao po svojem malom svijetu što se sastojao od pet soba jedne kosine i nekoliko kutija koje je mogao premještati s jednog mjesta na drugo. Njegov kompjutor je bio statičan, a Shakey je na njega bio priključen dugim komadom žice. Freddy je naprotiv bio statičan imao je ruku sa hvataljkom koja je visjela sa stropa dok se ispred njega nalazio stol koji se mogao pomicati u dva smjera pomoću električnih motora



Freddy je bio pokušaj da se u jednu cjelinu povežu umjetni vid strojna inteligen-



Desno dva najpoznatija filmska robota su C3PO i R2D2 iz *Rata zvijezda*. Iako prvi visoki može komunicirati na 3 milijuna načina a onaj drugi mali može popraviti interplanetarni brod u letu oni su ipak samo suvremene verzije Pata i Patašona u hipergalaktičkoj odjeći



Problem vide u industriji se često rješava vrlo jednostavnim sredstvima. Na slici se vidi kako se vertikalnom prugom svjetla osvjetljava neki mehanički dio što putuje na pokretnoj traci. TV kameri to daje dovoljnu indikaciju o položaju, veličini i obliku predmeta.

Stoljećima je bilo pokušaja da se stvore androidi.

Dolje desno: mademoiselle Cleo, bolničarke-instrumenterke koju je 1912. sagradio dr Herdner u pariškoj bolnici l'Hopital Brettonneise.

Dolje: kompjuterizirana čovjekolika lutka napravljena da studentima medicine pomogne u nastavi.

cija i robotska ruka, što je sve zajedno trebalo predstavljati radnika na pokretnoj traci. Bio je programiran za sastavljanje drvenih igračaka od dijelova složenih pred njim na stolu. Prije rada trebalo je Freddyju pokazati sve dijelove igračaka, u svim mogućim položajima (vidi str. 118-119), a potom uputiti ga kako da uhvati pojedini dio i kamo da ga postavi. Na kraju trebalo je još programom odrediti kojim redom da sastavlja igračku. Nakon ovih priprema, mora se priznati, Freddy se ponašao izrazito inteligentno.

Najprije bi pokušao iz hrpe dijelova odabrati one koje će trebati i te je stavljao na jednu stranu stola. Ako ih u gomili nije mogao prepoznati, pokušavao bi izvući pojedini dio i prepoznati ga, ali ako mu ni to ne bi uspjelo, razmahao bi se ljutito rukom i razbacao dijelove na sve strane stola.

Kao što smo vidjeli, suvremeni roboti su snažnih mišića, ali slabašne pameti. Njihove osjetila su oskudno razvijena. Na primjer kad se pred robotom-zavarivačem zaustavi pokretne trake s automobilskom karoserijom, on će odmah stati zavarivati marljivo i precizno. Ali ako se karoserija ne zaustavi pred njim, on će svejedno zavarivati, makar po zraku, ali jednako marljivo i precizno.

Očito je da se roboti dobro snalaze u preglednim situacijama te da su sposobni za obavljanje jednostavnih poslova. Ali zašto od umjetne inteligencije tražiti više nego što ona može pružiti. To je isto kao kad bismo kompjutoru postavljali pitanja poput onog „kako da postanem slavan i bogat". Sve kad bi na to pitanje i postojao neki određeni odgovor, vjerojatno ga ne bismo tražili od kompjutora.

Na str. 116-119 vidjeli smo kako je teško oponašati ljudski vid, pa čak kad bismo bili zadovoljni i s manjim zahtjevima. Jer jedno je naučiti stroj da raspoznaje nekoliko drvenih predmeta u laboratoriju pod idealnim uvjetima, a posve drugo napraviti uređaj koji će pouzdano raditi na svakom mjestu, bez obzira na vibracije, buku, prašinu i nepredvidivo osvjetljenje. Neočekivana sjena, mrlja masnoće ili odsjaj svjetla na komadu metala može lako zbuniti kompjutorov jednostavan i nesavršen vid.

Čak i pri idealnom osvjetljenju kompjutor još uvijek teško razabire što zapravo njegova kamera vidi. Pokazalo se da je najbolje predmet osvijetliti uskom prugom svjetla koje iz kosog smjera osvjetljava vidno polje kamere. Ali budući da se predmeti mogu na razne načine okrenuti, programeri ih moraju opisivati s grupama parametara dužine i opsega i iz različitih perspektiva (vidi str. 119).

Osjetilo dodira također postavlja kompjutoru velike zahtjeve. Nije teško opskrbiti robota hvataljkama s kontaktnim prekidačima, pneumatskim jastučićima ili nekim drugim senzorom za pritisak i dodir koji signalizira da li je hvataljka uhvatila neki predmet i s kolikom snagom. Teško je obraditi primljene podatke, što traži opsežne programe i troši mnogo vremena. Kompjutor mora stalno nadzirati senzore ako ne želi da mu promakne neka važna informacija poput one „pokušao sam uhvatiti zupčenik ali mi je ispao iz ruke".

Ako samo jedan kompjutor mora sve sam obaviti, proizvodna traka se mora toliko usporiti da više i ne služi svrsi. Ako bi se pak istodobno koristilo više kompjutora, susrećemo se s još neriješenim problemom paralelne obrade podataka (vidi str. 174).

Svakom industrijskom robotu treba objasniti njegove zadatke, baš kao što je to trebalo i Freddyju. Još uvijek je čovjek taj koji mora razraditi strategiju i program rada, te određivati „prvo učini ovo, a zatim ono". Ali ako od robota očekujemo da zaista ispune naša očekivanja, onda bi oni sami morali biti kadri da odlučuju o toku radnog procesa.

Shakey je predstavljao pokušaj da se u robota ugradi sposobnost donošenja odluka. Kretao se po svojem malom svijetu i zadaci koje je izvršavao bili su jednostavni: „Stavi ružičastu kutiju pokraj crvene u sobu broj 2". Shakey je prvo morao pretražiti svoju okolinu i ustanoviti gdje se kutije nalaze, a zatim je program (nazvan STRIPS) morao razraditi potrebne radne faze da bi se zadatak mogao izvršiti. U njemu su bile sadržane sve Shakeyjeve mogućnosti kretanja, traženje i nalaženje kutije, guranje kutija i prolaženje kroz vrata.

Nije teško ustanoviti da postoje dva načina na koja Shakey može izvršiti svoj zadatak.

Prvi način: odguraj ružičastu kutiju u sobu broj 2 i onda doguraj crvenu iz sobe broj 4.

Drugi način: odguraj crvenu kutiju u sobu broj 2, a zatim doguraj ružičastu kutiju.

Ako znamo gdje se nalazi crvena kutija, nije teško ustanoviti da je najprije treba odgurati u sobu 5, a nakon toga u sobu 2, itd. Čak i za tako jednostavan put treba mnogo kompjutorskih operacija. Osnovni problem je u tome, što kompjutor u svakoj fazi ima više mogućnosti. Što treba napraviti ako krene iz sobe 5? Shakey može ostati na mjestu ili otići u sobe 1, 2, 3 ili 4.

U sobi 4 on može odgurati crvenu kutiju na neko drugo mjesto ili kroz vrata u neku drugu sobu. Kad ružičasta kutija konačno bude u sobi 5, može ja prenijeti u sobe 1, 2, 3 i 4, ili ostaviti u sobi 5. Sad je već verbalno objašnjavanje postalo previše komplicirano, pa ja jednostavnije problem razmotriti na dijagramu desno.

Dijagram, dakako, ne daje sva moguća rješenja, jer ako bismo ispitali sve mogućnosti došlo bi do eksplozija kombinacija (vidi str. 175). Čak i mnogo jednostavnija pitanja, kao kako bih stigao do kolodvora, jer u 9 sati ide vlak u London, uključuje mnoštvo dedukcija iz svih mogućih znanja o automobilima, biciklima, autobusima, taksijima, vlakovima te o mjestima gdje je zgodno čekati i na onima gdje nije zgodno.

Međutim, ima nešto u našem mozgu što nam omogućava da probleme pred kojima je kompjutor bespomoćan, lako rješavamo. Da toga nema, već odavno bi nas nestalo s lica zemlje.

Pokušaj čije je vrijeme prošlo prije nego što je prikazan javnosti. Šezdesetih godina jedna ingeniozna američka kompanija demonstrirala je robota s jedinom funkcijom da izvuče svojih šest nogu iz kamiona, stane na tlo i vrati se u kamion.

Udaljana komunikacijske mreža na nacionalnoj i internacionalnoj razini povezuju se vodicem od optičkih vlakana ili preko satelita. Na slici desno vida se tahničan koji polažu akspenmantalni kabel između Londona i Birminghama. Premda su optička vlakna u kablu vrlo tanka moraju biti armirana i obavijena čvrstim plaštom kako bi izdržala surova uvjeta što vladaju izvan laboratorija. Na desnoj slici se vidi satelitska konkavna antena u Goonhilyju na britanskom kraju prekooceanske satelitska veza. Budući da su sateliti udaljeni 36 000 kilometara i opremljeni s relativno slabim odašiljačima, antena moraju biti vrlo velika da bi sabirale dovoljno energija za upotrebljiv prijenos podataka. Ali uskoro će tehnološki napredak omogućiti jako male antena da će se moći smjestiti na krov svakog ureda.





Cijela privreda i, mogli bismo reći, veći dio naše civilizacije, zaokupljen je izmjenom informacija. Šalju se pisma, računi, izvještaji, čitaju novine i knjige, gledaju TV programi i kazališne predstave, ali i ostavljaju poruke na kuhinjskom stolu. U ovom trenutku koriste se deseci različitih tehnologija za prenošenje informacija, od papira i olovke do komunikacijskih satelita.

Otkada su kompjutori ušli u svakodnevni život, sve se više povećavaju mogućnosti njihova povezivanja u informacijsku mrežu. Ona može sezati od nekoliko povezanih kompjutora u istoj zgradi, gdje je tok podataka najgušći, pa sve do povezivanja na velikim udaljenostima, kada je tok podataka doduše skromniji, ali zato može obići pola svijeta.

Mnogokorisničke mreže

Međusobno povezivanje osobnih kompjutora potaklo je ideju da se stvori elektronski ured o čemu se govori već nekoliko godina, a ponegdje je to već na pragu realizacije. Međutim, prije nego što se upustimo u detaljniju analizu tog problema, razmotrimo načine kako se kompjutori mogu međusobno povezati.

Pojednostavljeno gledano, postoje tri razine povezivanja. Prva je kad kompjutori šalju jedan drugome poruke preko serijskih linija (vidi str. 22-23), druga se sastoji u tome da više kompjutora koriste isti disk za izmjenu podataka uz normalnu brzinu pristupa, i treća razina, kad korisnici međusobno posuđuju memorije i procesore.

Najznačajnija osobina kompjutora je da svoje zadatke obavlja brže od čovjeka, a povezivanje na prvoj razini ozbiljno je ograničeno prijenosnom brzinom linija. Doduše, količina podataka koja stane na ekran video terminala može se telefonskom linijom prenijeti otprilike za 10 sekundi, što je dovoljno za kratke poruke ili hotelske rezervacije, ali sasvim nedostatno ako se želi prenijeti količine informacija po opsegu ravne sadržaju jedne knjige.

Ova razina izmjene informacija prikladna je za rutinsko komuniciranje u privredi, za kratke i formalne poruke, ali ne za opširnije poslovne izvještaje, liste faktura ili za financijske planove. Elektronika tu može samo djelomice pomoći, ali nije u stanju promijeniti osnovne načine komuniciranja.

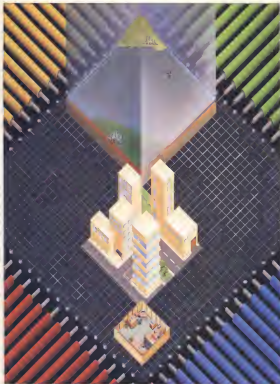
Ovdje možemo opet posegnuti za Zipfovim zakonom (str. 87), koji po prilici kaže da je količina informacije što ih dvoje ljudi izmjenjuje obrnuto proporcionalna udaljenosti među njima. Ta se tvrdnja sasvim slaže sa stvarnošću, jer troškovi komuniciranja naglo rastu s udaljenošću.

Druga razina — zajednički fajlovi na disku — predstavlja elektronički ekvivalent uredskom ormaru za spise. U uredima se doista tako radi: službenici uzimaju dokumente, rade na njima, pa ih onda vraćaju u arhivu, gdje su svakome dostupni.

Dakako, kompjuterizirani ured ima velikih prednosti pred svojim papirnim predšasnikom. U prvom redu svaki član kolektiva istodobno saznaje sve što je važno. Na primjer, ako službenici u nekom osiguravajućem zavodu koriste sustav terminala sa zajedničkim fajlovima, svi istodobno saznaju što se događa s kojim klijentom i drugo, pomoću kompjutora dobiju vrlo brzo sve informacije iz zajedničke baze podataka.

U takvim kompjutorskim konfiguracijama nužno je da se svakom korisniku omogući pristup zajedničkom disku na kojem se nalaze svi potrebni fajlovi. U klasičnom uredskom sustavu dvojica službenika ne mogu istodobno raditi na istom spisu, zbog jednostavnog razloga što je prvi već izvadio spis iz arhive pa ga drugi ne može naći. Kompjutorski ekvivalent sastoji se u blokiranju onog zapisa u bazi podataka (vidi str. 94-97) što ga je prvi korisnik već uzeo. Onaj koji kasnije pokuša tražiti isti fajl, na ekranu će naći natpis: zapis blokiran.

Na suprotnoj strani: tri razine komunikacijske mreže. Višekorisnička mreža u uredu (dolje), lokalna mreža u tvornici ili na sveučilištu (sredina) i daleka veza između gradova jedne države ili između dviju država povezana satelitima ili kablovima (gore).







Suvremeni ured (lijevo) i elektronički ured budućnosti (desno). Spisi se pohranjuju u kompjutor, s bilješke šalju putem mreže koja povezuje kompjutore. Informacije se primaju preko telefonskog modema (u prednjem planu). U uredu lišenom papira eliminiran je mučan, ni skokvalificirani posao koji se oslanja na mnogo rada s mnogo papira.

Za takve slučajeve u proizvodnji softvera postoje razni nazivi, kao 'otimanje fajlova' ili kolizija zapisa. Uostalom, termin i nije važan, važno je da softver koji se koristi u informacijskim mrežama automatski registrira i sprečava rad dvojice korisnika na istom fajlu, jer bi inače moglo doći do neugodnih posljedica. Taj problem jednostavno rješava softver Superfile, koji dopušta da nekoliko mikroračunala imaju pristup fajlovima istoga diska, ali onemogućava srazova. Danas, kad su u upotrebi diskovi kapaciteta do 30 megabajta, takvi sustavi mnogo obećavaju.

U osnovi postoje tri načina realizacije zajednička kompjutorske mreže. U višekorisničkom sustavu, svaka osoba ima svoj vlastiti kompjutor čvrsto vezan sa središnjim, glavnim kompjutorom, koji kontrolira disk. Svi ostali kompjutori prstenasto su vezani busom velikoga kapaciteta.

Prema prvoj organizacijskoj shemi svaki korisnik zauzima svoj dio memorije, a procesor radi naizmjenično. Tako se provodi paralelni rad na velikim računalima, što je svojedobno bilo najjeftinije, jer su procesori bili vrlo skupi, te svaki korisnik nije mogao imati svoj vlastiti. Procesori su bili dovoljno moćni da obiđu veliki broj korisnika i svakom pruže brzu uslugu, ali, na žalost, neki su konstruktori 8-bitnih višekorisničkih sustava slijepo kopirali ovu shemu i primijenili je i na velikim računalima. Nisu shvatili kako nema smisla prisiljavati više korisnika da dijele isti procesorski čip, vrijedan samo 5 dolara.

Glavni 8-bitni višekorisnički operacijski sustav izveden je iz CP/M sustava i nazvan MP/M. Svakom korisniku dodjeljuje oko 48K memorijskog prostora za izvođenje programa. To je dosta strogo ograničenje, jer procesor mora naizmjence obilaziti sve korisnike i za svakoga obavljati potreban posao. Procesor Z80 na primjer, sa svojih 8 bitova dovoljno je moćan za većinu uredskih poslova, ali ipak nije u stanju istodobno posluživati više korisnika.

Moćnija 16-bitna računala, u kojima radi familija Motorolinih procesora 68000, mnogo su pogodnija za takve poslove i dovoljno moćna da rade jednako kao i veliki kompjutorski sustavi. Često se služe istim operacijskim sustavom, Unixom na primjer, gdje nekoliko korisnika istodobno vrti različite programe na istom procesoru. To se naziva 'višenamjenskim' (multi-tasking) sustavom, a kod 16-bitnih jednokorisničkih računala naziva se 'Concurrent CP/M'.

Međutim, ako se koriste mnogo jeftinija 8-bitna računala, primjerenije je svakom korisniku dodijeliti vlastiti stroj, s potpunih 48K memorije, s ekranom i tastaturom, i omogućiti mu upotrebu istih diskova. To se obično naziva 'višeprocesorskim sustavom'. U praksi je najčešće rješenje da se pristup disku ostvaruje preko glavnog 'master' računala, čiji je jedini zadatak da posreduje između korisnika i diska, pa ga zato i nazivaju poslužiteljem fajlova (file-server). Glavno računalo, koje obično ima 16-bitni procesor, radi s odgovarajućim višeprocesorskim operacijskim sustavom. Jedan od najpoznatijih je CP/Net, koji svakom korisniku daje dojam da ima vlastiti CP/M. Postoji još i Turbodos, McNos, Hi-Net i još neki drugi, koji djeluju na isti način, i opet ima dvije metode po kojima se izvodi ovaj trik: po prvoj se nekoliko kompjutora, disk i glavno računalo stavljaju u istu kutiju, a svaki korisnik dobije svoj terminal. Po drugoj, svakom se korisniku dodjeljuje vlastito računalo povezano brzim komunikacijskim linijama s glavnim računalom i diskom.

Prednost prve metode je u tome što su zahtjevi na prenosne linije manji, jer se komunikacija između terminala i računala odvija u većim vremenskim intervalima i ne prenosi se odjednom

više znakove nego što ih stane na ekran. Kablovi su jeftini i mogu biti relativno dugački — do nekoliko stotina metara. Mana ove metode je u tome što svaki korisnik mora imati vlastiti terminal za prikaz i unošenje podataka, što poskupljuje cijeli sustav.

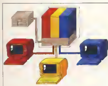
Prema drugoj metodi svaki korisnik dobije u svoj terminal vlastiti procesor, tako da zapravo posjeduje kompletno računalo. Time se štedi novac, ali treba osigurati brze prijenosne veze između svakog posebnog i glavnog računala. Radi kompatibilnosti sustava sva pojedinačna računala moraju biti od istoga proizvođača, premda neke informacijske mreže dozvoljavaju međusobno povezivanje različitih mikroračunala.

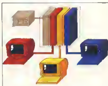
Treća, prstenasta shema koristi posebne međusklopove (interface) da bi se svaki kompjutor mogao priključiti na prsten, te vrlo brze prijenosne veze, obično putem koaksijalnih kablova. (Treba paziti da nas naziv 'prsten' ne zavara, jer se ovaj kabel nigdje ne spaja u zatvoreni krug.) Do nedavna su prstenaste konfiguracije bile prespore, pa ih nisu rado prihvaćali. Međutim, odlučno je što dozvoljavaju povezivanje kompjutora različitih proizvođača na glavno računalo, a to omogućuje korisnicima koji su dijelove sustava nabavljali iz različitih izvora, da ipak organiziraju elektronički ured.

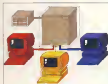
Meni se osobno čini da ljudi previše očekuju od ovakve konfiguracije, koja sama po sebi i nije loša, ali njome mora upravljati uigrana ekipa profesionalaca koja jedina može održavati tako osjetljivu strukturu. Držim da je primjerenija znanstvenicima u istraživačkim laboratorijama nego običnim korisnicima.

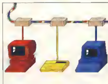
Mikrokompjutori se mogu povezati i tako da njihovi procesori postanu dijelovi istog velikog kompjutora. To znači da svi procesori rade zajednički i da svaki ima pristup cijeloj memoriji. Pretpostavimo, na primjer, da je vaš procesor besposlen, jer vi radite s programom za obradu teksta i pišući pismo nekom komitentu upravo gledate u strop i razmišljate da li da njega nazovete infamnim lažljivcem ili sebe žrtvom terminoloških nepreciznosti. Procesor susjednog računala, međutim, zauzet je sortiranjem velikog fajla i vaš procesor bez ikakvih smetnji priskače u pomoć.

Kasnije ćemo vidjeti (na str. 174-175) kako će se sljedeće generacija kompjutora služiti paralelno s više procesora, ali za to je potreban dalji napredak u tehnologiji proizvodnje čipova.









Postoji nekoliko načina da se više korisnika istodobno služi kompjutorom. Jedno računalo (gore lijevo) može posluživati disk i sve korisničke terminale prividno simultano. On zapravo poslužuje jednog po jednog, ali to se odvija tako brzo da se i ne primjećuje. Tako rade veliki kompjutorski sustavi (main frame), minikompjutori i jači 16-bitni mikrokompjutori.
Drugi način (gore desno), koji se često nalazi u običnim sustavima, omogućuje svakom korisniku pristup preko terminala do vlastitog kompjutora. Posebni kompjutor upravlja diskom do kojega svaki korisnik ima slobodan pristup.
Svaki korisnički terminal može imati i vlastiti procesor, povezan sa središnjim kompjutorom vrlo brzom prijenosnom vezom (dolje lijevo).
Četvrta varijanta (dolje desno) povezuje više samostalnih kompjutora, koji mogu biti i različite proizvodnje, pomoću kabela od optičkih vlakana.
Samostalni kompjutor može istodobno vrtjeti nekoliko programa samo za jednog korisnika (dolje).

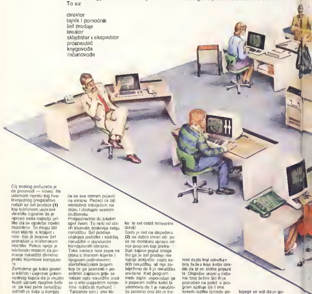


ELEKTRONIČKI URED

Vidjeli smo kako se mikrokompjutori mogu povezati da bi se njihovi korisnici služili istim fajlovima. Male radne organizacije su najpogodnije za primjenu ovakve tehnologije. U svijetu ih ima mnogo i način njihova poslovanja pogodan je za automatizaciju. Pogledajmo, na primjer, kako izgleda malo poduzeće za kreiranje, proizvodnju i prodaju ženskih šešira. Ima osam odjela i u svakom, zbog jednostavnosti, radi jedna osoba. To su:

direktor
tajnik i pomoćnik
šef prodaje
kreator
skladištar i ekspeditor
proizvođač
knjigovođa
računovođe

Svaki od ove osmorice ima mikroračunalo povezano s glavnim računalom i tvrdim diskom (hard disk) na kojemu je pohranjen cijeli arhiv poduzeća. Preko menadžera baze podataka svakome su dostupni svi poslovni podaci. Neki kompjutori imaju vlastite printere ili koju drugu vanjsku jedinicu, a neki koriste središnji printer priključen na glavno računalo. Pogledajmo kako izgleda jedan radni dan ove osmorice.



Cilj svakog poduzeća je da proizvodi — novac. Na udarnom mjestu tog hvalevrijednog pregalaštva nalazi se šef prodaje **(1)** koji telefonom uvjerava vlasnike trgovine da je upravo sada najbolja prilika da se opskrbe novim modelima. To mogu biti stari klijenti, a kojipnt i novi, čije je brojeve šef pronašao u telefonskom imeniku. Pokraj njega je telefonski modem za primanje narudžbi direktno preko klijentova kompjutora.

Zamislimo ga kako govori u telefon i uvjerava potencijalnog kupca da je mudro kupiti upravo njegove šešire, pa kad primi narudžbu odmah je šalje u kompjutorski sustav. U kompjutor će unijeti ime kupca, adresu, vrst trgovine itd. u standardnom obliku, što će se sve odmah pojaviti na ekranu. Podaci će biti istodobno zabilježeni na disku i dostupni svakom službeniku.

Pretpostavimo da telefon opet zvoni. To neki od stalnih klijenata dostavlja svoju narudžbu. Šef prodaje utipkava podatke i sadržaj narudžbe u standardni kompjutorski obrazac. Tako nastaje novi zapis na disku s imenom klijenta i njegovim jedinstvenim identifikacijskim brojem koji će ga povezati s posebnim zapisom gdje se nalaze opis narudžbe (radi se o vrlo uspješnim modelima „ružičasti mjehurić" i „Tarzanov san") ime klijenta i njegove adrese s opisom narudžbe, cijena s odobrenim popustom, kamo treba poslati pošiljku te svi ostali relevantni detalji.

Sada je red na dispečeru **(2)** da dobro otvori oči jer se na monitoru upravo odvija program koji pretražuje zapise poput onoga što ga je šef prodaje maloprije zaključio. Zapis sadrži narudžbu, ali nije zabilježeno da li je narudžba izvršena. Kad program nađe zapis, uspoređuje ga s popisom zaliha kako bi ustanovio da li je naručitelju poslano ono što je tražio.

Ne zalihi nedostaje tri modela Tarzanova sna, pa program mora postaviti novi zapis koji određuje broj šešira koje treba izraditi da bi se zalihe popunile. Dispečer unosi u računar broj šešira što ih je pronašao na polici, a program ispituje da li ima kakvih razlika između popisa zaliha i onoga što zaista stoji na policama skladišta. Nakon toga kompjutor stvara zapis iz kojega se vidi da je gospodi Jonson poslano 6 komada ružičastih mjehurića i 3 Tarzanova sna. Stvara se novi zapis s po-

datkom da gospođa Jonson treba još 3. Tarzanova šna a dispečer ispisuje na printeru bilješku o narudžbi i naljepnicu s adresom gospođe Jonson.

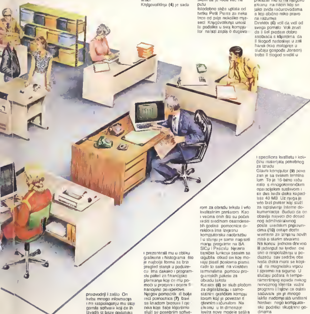
Ekspeditor **(3)** može sada zapakirati pošiljku. Program dodaje popisu koji se šalje u proizvodnju 3 narudžbe Tarzanova šna. Kad izrađeni šeširi dođu iz proizvodnje dodaju se popisu zaliha i ručno stavljaju na policu. Isti program također bilježi i materijal koji je upotrebljen za izradu šešira, a to treba računovođi i ostalim programima koji se bave na ručvanjem materijala. U bazu podataka sada ulazi zapis da su gospođi Jonson poslani naručeni artikli.

Knjigovotkinja **(4)** je sada zvijezda predstave. Njezin program pretražuje bazu podataka sa zapisima o narudžbama i bilježi da od danas gospođa Jonson duguje toliko i toliko za robu koja joj je poslana. Program priprema fakturu što će je poslati gospođi Jonson zajedno s bilješkom da je roba već na putu.

Istodobno stiže uplata od tvrtke Petit Parra za neka trice od prije nekoliko mjeseci. Knjigovotkinja unosi te podatke u svoj kompjutor, nalazi zapis o dugovanjima i briše pristiglu sumu.

Od vremena do vremena računovođa **(5)** želi vidjeti što se događa — što je prodano, što ima na skladištu i koliko novca je uplaćeno. Njegov će program prošvrljati bazom podataka, naći traženo i prikazati mu to na njegovu ekranu na način koji se jako sviđa računovođama, a koji obično nitko pravo ne razumije.

Direktor **(6)** voli da vidi od svega pomalo. Voli znati da li šef prodaje dobro saobraća s klijentima, da li štogod nedostaje u zalihama (kao maloprije u slučaju gospođe Jonson), treba li štogod srediti u



proizvodnji i zašto. On treba mnogo informacija i na raspolaganju mu stoji gomila softvera koji će ih izvaditi iz baze podataka i prezentirati mu u obliku grafikona i histograma, što je najbolja forma za brzi pregled stanja u poduzeću. Ima dakako i programski paket za financijsko planiranje koji će mu pomoći u procjeni i ocjeni financijske perspektive.

Njegov pomoćnik, ili bolje reći pomoćnica **(7)**, bavi se izradom brošura i cjenika koje šalje klijentima, služi se posebnim softverom za obradu teksta i vrlo kvalitetnim printerom. Kao i većina onih što su počeli raditi sredinom osamdesetih godina, pomoćnica direktora ima izvrsnu kompjutorsku naobrazbu i u stanju je sama napisati manje programe na BASICu i Pascalu. Njezina tajnička funkcija sasvim se izgubila otkad svi koji moraju pisati poslovna pisma rade to sami na vlastitim terminalima pomoću programskih paketa za obradu teksta.

Kreator **(8)** se služi pločom za digitalizaciju i samostalnim grafičkim kompjutorom koji je povezan s glavnim računalom. Na ekranu u tri dimenzije kreira nove modele šešira i specificira kvalitetu i količinu materijala potrebnog za izradu.

Glavni kompjutor **(9)** povezan je sa svakim terminalom. To je 16-bitno računalo s mnogokorisničkim operacijskim sustavom i sa dva tvrda diska kapaciteta 40 MB. Uz njega je vrlo brzi printer koji služi za ispisivanje interne dokumentacije. Budući da on obavlja najveći dio dosadnog administrativnog posla, uredskim pripravnicima **(10)** ostaje dosta vremena za izmjenu novih misli o starim stvarima.

Na koncu, jednom dnevno ili jedanput na tjedan, ovisno o raspoloženju u poduzeću, sav sadržaj oba tvrda diska mora se kopirati na magnetsku vrpcu i spremiti na sigurno. U slučaju požara ili temperamentnijeg ispada nekog nervoznog klijenta, važni programi i fajlovi će ostati sačuvani, jer je mnogo lakše nadomjestiti uništeni hardver nego kompjutorske podatke skupljane godinama.

DALJINSKE MREŽE

Klasična telefonska mreža u konfiguraciji zvijezde Pretplatnički telefoni povezani su s lokalnom centralom lokalne centrale sa glavnom a glavne s međunarodnim centralama Takva mreža nije najsret nije rješenje pogotovo za povezivanje kompjutora ali mora dalje funkcionirati jer je u nju uloženo previše novaca da bi se mogla mijenjati

Što se više kompjutori rasprostiru po svijetu to više preuzimaju funkciju telefona teleksa i naposa poštanskih usluga U daljem razvoju služit će i za slanje i primanja faktura računa otpremnica saopćenja i izvještaja

Poštanske službe su polagane Obavljaju ručno ono što bi se moglo uz pomoć elektronike obaviti brzo Bilo kakvu poštansku pošiljku — svejedno da li se radi o nekom računu ljubavnom pismu ili paru čarapa — treba staviti u omotnicu i adresirati Povijest pošta nas uči da su i pisma s adresom N N Ulica Seljačkog kralja Zagorja ipak stigla na pravu adresu u ulicu Matije Gupca u Stubici Bilo bi dobro kad bi i kompjutor mogao postupati na isti neformalni način jer pošta mora koji put uručiti pošiljku makar je na njoj nepotpuna adrasa Dobra poštanska informacijska sluzba morala bi voditi evidenciju i o promjenama adresa a naslovljavanje se na bi moralo obavljati samo prema mjestu stanovanja već i prema zanimanju ili interesima Tako biste na primjer mogli slati obavijesti svim vlasnicima Spectrum kompjutora u gradu ili svim veterinarima u jednoj pokrajini Adresa na bi smjela biti samo zbir suhih podataka o gradu ulici i kućnom broju adresanta već bi morala kazivati i nešto više o osobi na koju se odnosi

Telefonska mreža predstavlja modal svake elektroničke mreže U već davno prihvaćenom telefonskom sustavu s konfiguracijom zvijezda na svaku se lokalnu centralu može priključiti do 9999 pretplatnika (to je zato što je pojedini telefonski broj limitiran na 4 digita) Lokalne centrale povezana su u zvijezdu i priključene na tranzitne ili međugradske centrale a ove pak istim načinom na međunarodne

Nije teško shvatiti kako se pronalazi pojedini pretplatnik prvo se bira pozivni broj zemlje zatim broj područja ili grada pa broj lokalne centrale i konačno broj pretplatnika što će vas glas preko dviju zica dovesti do osobe koju tražite Treba dakle okrenuti desatak ili našto više znamenki da se izabere bilo koji od nekoliko stotina milijuna telefonskih priključaka povezanih u svjetsku telefonsku mrežu To ima mnogo prednosti sva elektronička telefonska 'pamet je koncentrirana u centralama koje PTT može lako održavati i štititi (telefonske centrale su prirodne mete terorista i diverzanata) telefonski aparat je jednostavan i jeftin a raspored zvijezde najbolje odgovara telefonskom prometu Iako se sa svakog telefonskog aparata može pozvati bilo koji telefonski pretplatnik na svijetu ipak se najviša telefonskih razgovora obavi u gradskom i područnom prometu Gustoća telefonskih poziva uglavnom slijedi Zipfov zakon (vidi str 87)

Konfiguracija zvijezde' ima međutim i svojih mana Najgore je što zahtijeva ogromnu količinu vodova jer svaki pretplatnik mora biti individualno vezan na centralu Nije jednostavno priključiti novoga pretplatnika jer za njega u centrali treba instalirati dodatne sklopove i povući par novih zica sva do njegove kuća

Do sada smo govorili o komuniciranju putem telefona što znači zvučnim signalima Ali ista terminologija se može primijeniti i na kompjutore jer se kompjutorski digitalni podaci mogu pretvoriti u zvučne signale a iz njih opet u digitalne (vidi str 120-123)

Vec više godina kompjutori se povezuju preko kanala za veze na velikim udaljenostima a odnedavna se važu u mrežu oko centralnog procesora dok je više centralnih procesora povezano međusobno u širu mrežu Jedna od najpoznatijih mreža ove vrsti je APRA (Advanced Projects Research Agency) Da bi se ušlo u mrežu treba se priključiti na određeni procesor i poznavati komunikacijske protokole što je sve vrlo komplicirano i vrlo skupo dostupno uglavnom sveučilištima i vojsci Podaci su sakupljeni u pakete standardne dužine koji se šalju linijom jedan po jedan, poput željezničkih vagona

Pošte u razvijenim zemljama planiraju izgradnju informacijskih kanala pomoću vlastitih kompjutora premda je pristup ovim mrežama općenito skup i nespretan te ni izdelaka tako fleksibilan kao klasična poštanska služba Onda se netko dosjetio (naravno u Americi) da bi se moglo služiti improviziranim komunikacijskim mrežama pomoću telefona

Kao što smo gore vidjeli prijenos podataka telefonom je spor nepouzdan i neelastičan Nešto se može učiniti ako se napiše softver koji će na sebe preuzeti biranje brojeva i provjeravanje da li je uspostavljena veza s pravim

Na suprotnoj strani
Dolje neki proizvođači eksperimentiraju s kombinacijom telefona i TV ekrana za primanje Prestel i drugih video teksi sustava.

Dolje desno Homelink sustav na kojem pretplatnik ispituje stanje svoga bankovnog računa Preko sličnih terminala postavljenih u trgovinama može se direktno prebaciti novac s bankovnog računa kupca na račun trgovine





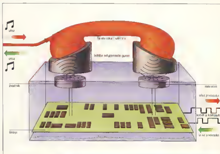
Gore dva primjera elektronske raspodjele informacija. Prva slika prikazuje vodič po restoranima britanske kompanije Telecom Prestel, a druga Celaxov program u BASICu za kompjutorske entuzijaste koji emitira mreža BBC

kompjutorom (ili pravim osobom) Britanska kompanija Telecom Prestel je, na primjer, razradila mrežu za distribuciju podataka koja je dostupna svakome tko je na nju priključen, ali je slika na monitoru bila tako gruba i tako ju je bilo teško fiksirati da je cijeli projekt doživio neuspjeh

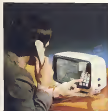
Telecom je tada ponudio Gateway službu koja je povezivala telefonskom linijom svaki kompjutor s bilo kojim drugim preko Telecomova kompjutora. Ali i to je bilo skupo i nepriklađno, pa se tako sve do danas nije našlo pravo rješenje te se primjena kompjutora u masovnoj komunikaciji nije pravo niti počela ostvarivati

Jednostavno rečeno, željeli bismo da svaki od milijuna kompjutora u svijetu može odaslati svoje podatke bilo kojem drugom kompjutoru na bilo kojem kraju svijeta. To su dakako veliki zahtjevi i zato najprije treba razmotriti mogućnost povezivanja ograničenog broja računala. Kao i uvijek, postoje dva načina: prvi računa s vezama organiziranim poput zvjezdaste telefonske mreže. Svaki pojedini terminal priključen je na centralu, svaka centrala na višu centralu itd. Drugi način povezivanja oslanja se na princip busa (vidi str. 22-23) ili sabirnice, gdje su svi korisnici priključeni na opći tok podataka, a primaju samo one koji su na njih adresirani

Dugoročno gledano, prvi način vjerojatno nije



Desno telefonski modem prima od kompjutora digitalne signale, pretvara ih u zvučne različite frekvencije i šalje dalje telefonskim linijama. Jednostavan amaterski modem poput ovoga koristi slušalicu standardne izvedbe koja se uleže u jastučiće od pjenaste gume. Profesionalne izvedbe modema nemaju zvučnika i mikrofona, već se priključuju direktno na telefonsku liniju





Prije nešto više od deset godina britanska PTT je eksperimentirala s telesastanicima. Studiji su bili organizirani po većim gradovima i poslovni ljudi koji su se trebali sastati odlazili bi u najbliži studio gdje ih je povezivala televizijska slika i strojevi za kopiranje dokumenata. Eksperiment međutim nije imao većeg uspjeha vjerojatno zato što je bio ispred svojega vremena iako će se u skoroj budućnosti slične mogućnosti ostvariti pomoću kompjutorskih mreža.

dobar jer je previše nespretan i ograničen a mreža centrala prekomplicirana. Osim toga postavlja se pitanje čemu organizirati centrale za uključivanje kompjutora kad je sve što treba za uključivanje ionako ugrađeno u svako računalo. Druga metoda postavlja jednostavnije zahtjeve na tehničku infrastrukturu jer po tom sustavu se svi signali dovode do svakog terminala, a za to je najzgodnije sredstvo vodič od optičkih vlakana.

Vodič od optičkih vlakana povezuje jednog korisnika s drugim i sve ostale zajedno i tako tvori komunikacijsku mrežu. To je slično kao kod radija sve što se emitira u principu može svatko primiti pa stoga nije potrebno prespajanje kao u telefonskim centralama, jer se cijeli proces selekcije podataka odvija u korisnikovu terminalu. Sama mreža bila bi vrlo ekonomična ako za primjer uzmemo britanske otoke ni jedan pretplatnik ne bi bio udaljeniji od drugog više od 1600 kilometara, što znači da bi signalu trebalo pola milisekunde da stigne do najudaljenijeg pretplatnika i natrag.

Dok nitko ništa ne govori, nema prijenosa. Ali u času kad terminal A želi poslati poruku terminalu B (svaki ima identifikacijski broj nešto poput telefonskog broja) istog trenutka, na svim linijama nastaje živost. Najprije A treba privući pažnju master jedinice koja prima oba broja i provjerava paritet (vidi str. 12-13) kako bi bilo sigurno da je poruka valjana i da brojke eventualno nisu izmiješane i šalje poziv B-u. A i ostali na liniji slušaju ali pošto je poziv upućen B-u, svi ostali šute. Ako je terminal B uključen i ispravan šalje svoj identifikacijski broj kao potvrdu što master opet provjerava. Ukoliko je odziv terminala B u redu A može početi emitiranje poruke i kada završi, šalje signal 'gotovo', koji master prenosi ostalim terminalima i daje znak da je mreža slobodna.

Pažljivi čitatelj se može zapitati što se događa ako A i B istodobno pošalju zahtjev za slanje poruke. Dakako master tada prima dva superponirana signala provjerava ih nalazi da su oba neispravna i ne odziva se. Kako ni A niti B nisu dobili potvrdu o primitku signala oba pokušavaju ponovo uspostaviti vezu. Da bi se spriječile dalje kolizije oba terminala čekaju neko vrijeme što ovisi o procjeni opterećenosti mreže pa pokušavaju još jednom. Prema zakonu vjerojatnosti ovaj put neće doći do kolizije, pa onaj koji prvi privuče pažnju mastera prisiljava drugoga da šuti dok sâm ne završi.

Taj je sustav prvi put razvijen za kompjutorsku mrežu 'Aloha' (na polinezijskom aloha znači pozdrav kao naše 'zdravo') sredinom sedamdesetih godina na Havajima. Aloha je upotrebljavao obične vodove a ne one od optičkih vlakana ali je princip bio isti. Moglo bi se pomisliti da se najviše vremena trošilo na rješavanje kolizija signala ali se pokazalo da su pozivni signali tako kratki, da je vjerojatnost kolizije minimalna. Mreža može raditi s 80% svog teoretskog kapaciteta, ako se emitiranja obavljaju po redu i poruke šalju bez pauze.

Postoji mogućnost da se ciklički proziva svaki terminal i provjerava da li ima ili nema poruku. Međutim vrijeme bi se tada trošilo i na neaktivne terminale i bolje je da master vodi evidenciju o stanju u mreži o tome tko je ukopčan a tko nije.

Takva mreža može imati vrlo veliki kapacitet prijenosa podataka. Vodiči od optičkog vlakna mogu prenijeti po prilici 300Mb/s (megabita na sekundu) što znači da se sa realnim faktorom opterećenja od 80% može prenijeti 240Mb/s ili 30MB/s (megabajta na sekundu). Da bi se stekao pravi dojam o veličini ovih brojaka treba ih usporediti s najsporijom karikom u lancu prijenosa informacija s našim očima za čitanje i prstima za tipkanje. Malo je onih koji mogu pročitati i usvojiti više od 5 000 riječi dnevno a ima pisaca koji mogu napisati i 1 000 riječi dnevno ali ih sigurno nadmašuju profesionalni korespondenti i sastavljači propagandnih materijala. Potencijalni korisnik mreže može biti zadovoljan s 5 000 riječi dnevno ili 30 000 znakova (riječ ima u prosjeku 6 znakova) tako da bi jedan jedini vodič od optičkog vlakna rastegnut preko zemaljske kugle mogao osigurati komunikacijske potrebe za 90 milijuna ljudi.

Kako u praksi ljudi rijetko šalju poruke iz Londona na otočje Orkney (vidi Zipfov zakon str 87) bilo bi bolje ograničiti mreže na uža geografska područja. Na većim udaljenostima informacijske mreže će vjerojatno koristiti opto-kablove spuštene na dno oceana satelite ili mikrovalne radio odašiljače. Takvi kanali visoka kapaciteta bit će vrlo skupi pa će kanali funkcionirati na ponešto drugačiji način. Podaci će se vjerojatno sakupljati i prosljeđivati u paketima kao što se i sada radi u velikim kompjutorskim sustavima.

Prednost Aloha sustava je u tome što adrese nisu vezane uz geografske pojmove. Međutim, kad nazovemo neki telefonski broj, mi zapravo nazivamo određenu kuću ili ured bežívotnu stvar s kojom se ne može razgovarati ali se na-



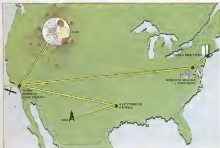




Gore lijevo budućnost je već počela. Scena iz jedne londonske banke gdje su službenici povezani telefonima i kompjutorima s ostalim članovima bankovne mreže. Na pritisak prsta kompjutori izračunavaju kolikí će se profit ostvariti ako se 30 milijuna maraka proda za sto procenata skuplje.

Gore u sredini liječnik ispituje pacijentov rentgenogram zajedno s kolegom koji se možda nalazi u drugom gradu. Pred obojicom je ista slika prenesena kompjutorom.

Gore desno u laboratorijima tvrtke Bell vrše se pokusi sa sustavom EPCOT. Pomoću ekrana osjetljivog na dodir slike se prenose video vezama do drugog udaljenog terminala.



Lijevo primjer kako mikro kompjutor i daljinska mreža mogu služiti piscu filmskih scenarija. Vjerujemo da piscu godi mir i osamljenost planina. Scenarij na kojem upravo radi bavi se piljevim intrigama što ih prave ljudi s naftonosnih polja. U ovom trenutku zanimaju ga neki tehnički detalji, pa je stupio u vezu s jednim prijateljem iz Dallasa koji se u to razumije. Malo kasnije će zatrebati još neke podatke sa njujorške burze kao i jednu konzultaciju s kongresnom bibliotekom u Washingtonu. Konačna verzija scenarija upućena je preko kompjutorske mreže (dakako sa zakašnjenjem) producentu u Los Angelesu, a ovaj ju je proslijedio putem satelitske veze partneru koji se nalazi na poslovnom putu u Tokiju.

damo da se osoba koju tražimo nalazi u toj kući ili uredu, što može biti istina, ali i ne mora.

Ono što želimo je da pozovemo određenu osobu, makar točno i ne znamo gdje se ona nalazi. U telefonskom sustavu to je vrlo teško izvesti, jer bi svatko, prije nego što izađe iz kuće, morao na kućnoj centrali ostaviti poruku kamo je krenuo. Ali kao što često biva u životu, može se dogoditi da ste ostavili poruku kako ćete popodne provesti kod Sanje, a naknadno ste se predomislili i krenuli još do Nataše, pa bi opet trebalo ostaviti novu poruku da ste ... i stvari bi se toliko zakomplicirale, pogotovo u telefonskom "zvijezda" sustavu, da ih nitko više ne bi mogao raspetljati.

Naprotiv, u Aloha sustavu, gdje se sva logika nalazi u terminalima, adresiranje je vrlo jednostavno. Svaki terminal ima fiksni broj koji odgovara njegovoj adresi, ali mu se mogu pridružiti i neki drugi brojevi. Tako, kad stignete Sanji, jednostavno programirate njezin terminal da prihvaća pozive i za vas. To isto učinite i kod Nataše i Brankina poruka će nepogrešivo slijediti vaš put, a kompjuterizirani računi pronaći će dužnike gdje god oni bili.

Sustav adresiranja može se proširiti na interesne grupe, pa tako možete primati sve vijesti koje se odnose na vašu amatersku družinu, možete se pretplatiti na časopise, novine, revije i biltene, koji se distribuiraju u digitalnom obliku preko kompjutora. Sve će vas to slijediti kamo god krenuli. Mogućnosti su očito nevjerojatne.

Optička vlakna

Tokom osamdesetih godina bit ćemo svjedoci nagiog prelaska s bakarnih vodiča i radio veza (direktnih ili satelitskih) na vodiče od optičkog vlakna. Tih je vodova položeno već na tisuće kilometara u Evropi i Americi, a prvi podmorski

kabel koji će povezivati Japan i Havaje bit će položen do 1988 godine

Vodiči od optičkog vlakna sastoje se od tanahnih staklenih niti kroz koje se odgovarajućom brzinom šire svjetlosni bljeskovi Impulsi se generiraju laserom ili laserskom diodom a sadrže digitalno kodirane informacije kompjutorskih podataka zvuka ili TV signala

Staklo je tako čisti medij da se signal kroz njega širi do udaljenosti od 35 kilometara prije nego što ga treba pojačavati (običan bakreni vodič traži pojačalo na udaljenosti od jednog kilometra pa čak i manje) Kapacitet prijenosa vodičem od optičkih vlakana ograničen je na 300 Mb/s zbog svojstava poluvodiča koji se koriste za generiranje i detekciju svjetlosnih impulsa Kako se razvija tehnologija poluvodiča tako će se i kapacitet postojećih veza povećavati Osim toga kada milijuni kilometara bakrenih vodova širom svijeta budu zamijenjeni staklenim vodovima to će znatno povećati svjetske zalihe tog dragocjenog crvenog metala

Lijevo geostacionarni re lejni satelit za prijenos po daleka koji leodi u fiksnoj točki iznad ekvatora Krila su mu prekrivene solarnim celijama iz kojih crpi elek tričnu energiju a konkavne antene usmjeravaju radio signale u smjeru određe nih područja zemaljske kugle

Gore lijevo stakleni šlap sirovine za proizvodnju vodiča od optičkog vlakna izvlači se iz peći za taljenje u kirurški sterilnim uvjetima

Gore sredina tenka poput kose nit optičkog vlakna izvlači se iz staklenog šlapa tlaućo pula čistijeg sastave od običnog stakla

Gore desno informacije se šalju kroz optičko vlakno pomoću laserskih impulsa

Desno za prijenos podataka na kraće udaljenosti upotrebljavaju se optička vlakna većeg promjera sastavljena od dvije vrste stakla jezgra kojom putuju signali je od jedne vrste stakla a omotač čiji je zadatak da gubitke po sredstvom optičke refrakcije smanji na najmanju mjeru od druge Ovaj snimak na kojem se lijepo vide dvije vrste stakla snimljen je u Bellovim laboratorijima





GOLEME BAZE PODATAKA

Vrlo je vjerojatno da će prodiranje kompjutora u domove i urede te uspostavljanje brzih komunikacijskih linija imati dramatske posljedice na pohranu i pristup svim vrstama podataka Danas je situacija takva da neke nepoznate činjenice morate potražiti u knjigama Prolistat ćete opću enciklopediju pa ako to ne pomogne, posegnut ćete za nekom specijaliziranom enciklopedijom a nakon toga za stručnim priručnicima Mnogi podaci se danas brzo mijenjaju poput željezničkih voznih redova, pa zato treba konzultirati knjige uz koje se stalno objavljuju dopune A možda želite saznati kako treba reklamirati pseću hranu na Filipinima ili vas zanima tko su dobavljači lijevanih cijevi u Boliviji? U tom slučaju morat ćete prelistati mnoge trgovačke priručnike

Sve je te informacije morao netko prikupiti i objaviti a na vama je da odaberete pravo izdanje Ali postoje još mnogi drugi podaci koje je vrijedno znati a nisu sakupljeni niti objavljeni na primjer kolika je cijena dionica honkonških kompanije što posluju s Kinom ili kakvi su poslovni rezultati zapadnonjemačkih proizvođača kompjutora

Suvremene tehničke mogućnosti neizbježno će revolucionirati svjetski informacijski sustav Trend objavljivanja vrućih informacija (kretanje cijena dionica sportski rezultati zabavne priredbe redovi vožnje itd) sve je prisutniji u svijetu

Taj se proces nastavlja Ljudi uviđaju da se mnoge informacije što su do sada objavljivane na papiru mogu mnogo lakše distribuirati u formi podataka za mikrokompjutore ali da bi to postala stvarnost, treba cijelu svjetsku zalihu informacija prevesti u digitalni oblik a to je zaista velik pothvat

U drugoj fazi razvoja može se očekivati da će se proces pomoću inteligentnog softvera potpuno automatizirati Kada svi svjetski izvori informacija budu ujedinjeni u golemim bankama podataka, moći ćete narediti svom osobnom kompjutoru 'Nađi mi sve što znaš o slikaru Turneru' ili 'Pripremi mi izvještaj o svjetskoj proizvodnji jute ' Pametna mala naprava poslušno će se baciti na posao konzultirati cijeli niz drugostepenih baza podataka gdje su katalogizirani glavni izvori informacija i postaviti im pitanja reci mi ovo pronađi ono a kad dobije odgovore, sredit će ih da budu što pregledniji, otisnuti na printaru i usput formirati mini-bazu podataka u vlastitoj memoriji

Ovakvi programi daleko će nadmašiti suvremene menadžere baze podataka Morat će biti vrlo inteligentni sposobni da shvate o čemu se radi primijeniti najbolji način da sakupe što više podataka o zadanoj temi te da onda jurnu u svijet informacija poput psa koji gospodaru spremno donosi bačeni štap

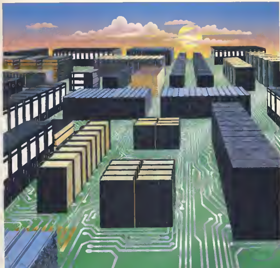
Posebno će se tražiti uređaji za automatsko čitanje jer će se svo znanje akumulirano u svijetu prevesti u elektronski oblik U Britanskoj državnoj knjižnici ima oko 13 milijuna knjiga od kojih svaka sadrži prosječno 60 000 riječi ili 360 000 znakova Ako u svijetu postoji još dvadesetak takvih knjižnica znači da se mora ispisati oko 93 bilijuna (10^{12}) znakova Dodamo li tome još i dnevne listove a njih se svakodnevno u svijetu tiska preko 5 000 to je daljih 100 000 riječi ili 600 000 znakova Taj broj treba još povisiti za svu znanstvenu i tehničku literaturu u posljednjih 100 godina što iznosi još 54 bilijuna bajtova koje treba prevesti u ASCII format Tako ogroman posao mogu obaviti samo strojevi a takvih danas još nema ili barem nisu dovoljno moćni Dakako da će za taj posao trebati programi — prevodioci

Postoji međutim tamna strana svega ovoga Znanje je moć a slobodno raspolaganje njime nije u svačijem interesu Primjer su državni arhivi Španjolske gdje su pohranjeni podaci o potonulim galijama Dok se o njima nije ništa znalo lovci na blago s potonulih brodova nisu se javljali a ozbiljni su istraživači mogli mirno raditi Malo je ljudi koji znaju kako se treba ponašati u arhivima i kako treba postupati s dragocjenim starim rukopisima a sada nakon što je nestalo tajne taj je arhiv postao ciljem mnogih lovaca na blago Ima još mnogo nepročitanih dokumenata koji leže duboko u policama i nitko ne zna da su tamo Ali kad bi svi dokumenti na pritisak tastera postali svakome dostupni trebalo bi ipak na neki način ograničiti pristup banki podataka što bi se vjerojatno postiglo šifriranjem Samo provjereni ljudi posjedovali bi ključ šifre i bio bi to dosta tužan svršetak stoljetne tradicije akademskih sloboda

Kodiranje

Elektronske podatke je vrlo lako kopirati što ponekad predstavlja ozbiljan nedostatak Suprotno tome dokumenti na papiru imaju nekih prednosti kao što je trajnost i u nekim slučajevima neponovljivost Na primjer novčanice je gotovo nemoguće falsificirati S druge pak strane ne postoji elektronički ekvivalent novčanicama što bi se slale od kompjutora do kompjutora a da ih se ne bi moglo kopirati Postoje dakako osiguranja koja sprečavaju miješanje trećih osoba u komunikacije između banaka i klijenata a osim toga obje strane vode dnevnik poruka pa uvijek mogu provjeriti ispravnost izvršenja

Kompjutoru ne bi bilo teško tako temeljito izmiješati tekstove i brojeve te bi i nekoliko godina trebalo raditi uz pomoć vrlo moćnih strojeva da se probija šifre Međutim postoji jedna zanimljivija alternativa koje su se nedavno dosjetili matematičari Oslanja se na neka matematičke postupke koje je vrlo lako provesti u jednom smjeru ali vrlo teško u suprotnom To je na primjer traženje prim brojeva u većim brojevima (200-300 znamenki) što zahtijeva ogroman posao dok je u obratnom smjeru ako je



poznat prim broj vrlo jednostavno množenjem dobiti argument rastavljanja. Taj princip može poslužiti za sustav kodiranja u kojemu šifra može biti javna (to je ono što začuđuje) i šifrirati može svatko, ali dešifrirati samo onaj što je šifru izmislio. Ako se ovakve dvije šifre međusobno spoje, obje strane — na primjer banka i klijent — sigurne su da poruke može čitati samo onaj koji ima na to pravo.





REVOLUCIJA MIŠLJENJA

Jedan od mnogih razloga da kompjutori izazivaju otpor kod obična čovjeka je u tome što traže neke radikalne pomake u svijesti i oslobađanje od zdravorazumskog pogleda na znanost naslijeđenog od samozadovoljnih znanstvenika prošlog stoljeca. Industrija je te pomake usvojila nesvjesno, dok u svakodnevnom životu ljudi i dalje osjecaju odbojnost prema kompjutorima, mada tome ne znaju pravih razloga. Teškoće koje su pratile i još uvijek prate kompjuterizaciju slične su potresima što su revolucionirali matematiku u prvoj potovici ovoga stoljeca.

Zabunu su tada izazvale tri pitanja što ih je 1900. godine David Hilbert uputio matematičkom svijetu: da li je matematika *potpuna*, zato jer se svaka tvrdnja u njoj može dokazati ili osporiti? Da li je matematika *konzistentna*, jer se u njoj valjanim postupkom ne može doći do krive tvrdnje, i da li je matematika *odlučive* ako u njoj postoji, barem u principu, neki konačni postupak kojim se može istražiti istinitost svake tvrdnje?

Na prva dva pitanja odgovono je 1929. godina mladi madžarski matematičar Kurt Godel služeći se metodom koja se u to vrijeme mnogima činila ludom — kodiranjem matematičkih pravila i formule brojevima. Pridruživši svakoj formuli odgovarajući kodni broj, dokazao je da u svakom aksiomatskom sustavu uvijek postoje dodatni kodni brojevi, tj. tvrdnje koje se ne mogu izvesti iz onih što već postoje. Drugim riječima, uvijek će postojati tvrdnje u svakom logičkom sustavu (ne samo u matematici) koje se ne mogu ni dokazati niti oboriti onim što već postoji. Također je dokazao da se matematika ne može smatrati skladnom, a da se ne uvedu dodani aksiomi izvana.

Sve je to bilo loše za matematiku, ali dobro za teoretsku podlogu budućem razvitku kompjutora, jer je Godel stvorio pukotinu u uvjerenju da su matematička pravila i formule jedno, a brojevi koji iz njih proizlaze — drugo. On je jednostavno ustvrdio da je svako matematičko pravilo ujedno i broj. Tada je to izgledalo neobično, ali danas je prihvaćeno kao gotova činjenica. Kada se, na primjer, napiše u BASICu

```
100 IF A>34 THEN C=COS(K) ELSE C=SIN(K)
```

kompjutor u tome vidi samo brojeve u heksadecimalnom sustavu

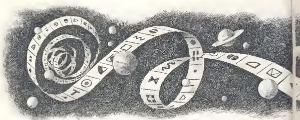
```
FF 0D 60 64 00 8B 20 41 EF 0F 22 20 CF 20 43
F0 FF 8C 28 4B 29 20 3A A2 20 43 F0 FF 89 28
4B 20 00 00
```

Svaka formula odgovara dakle nekom heksadecimalnom broju i nema zapreke da se izračuna suma tih brojeva ili da ih se složi po nakom redu.

Taj novi odnos prema simbolima bio je jedan od ključnih preduvjeta za nastanak kompjutora.

Drugi važan preduvjet za nastanak kompjutora bilo je napuštanje ideje o stroju za računanje. Čim je čovjek naucio računati (a to je bio izuzetan intelektualni napredak) pokušao je automatizirati aritmetičke postupke. Paralelno s razvojem mehanike nastajali su sve kompliciraniji strojevi za računanje, ali svako dalje poboljšanje tražilo je i potpuno nove konstrukcije.

Sredinom tridesetih godina našega stoljeca Alan Turing, plahi mladi matematičar na Kraljevskom koledžu u Cambridgeu, načeo je još jedan problem. Na mjestu gdje se Gödel našao pred zidom, Turing je tvrdoglavo nastavio. On je ustvrdio: ako postoji metoda za rješenje bilo kojeg teorema, matematičar (ili računač, kako se tada nazivao) treba samo dosljedno primjenjivati pravila, pa će svaki teorem biti rješiv. Išao je i dalje te postavio pitanje što će se dogoditi ako živog matematičara zamijeni računski





stroj? Stroj može biti tisuću puta glupiji od ljudskoga bića ali ako se pridržava istih pravila na kraju će doći do istog rezultata

Turing je smatrao da bi neki matematički teorem pretvoren u niz simbola stroj trebao obraditi na nepogrešivi mehanički način i na koncu ustanoviti da li je dotični teorem dokaziv ili nije. Drugim riječima, trebao je raditi poput kompjutora samo što tada nikome nije padala na pamet takva usporedba

Vjerojatno ni ne znajući da vuče dalekosežne poteze Turing je iz svoje zamišljene mehaničke konstrukcije izbacio mnoštvo zupčanika osovina i poluga što su ih raniji konstruktori ugrađivali u svoje strojeve jer je njegov uređaj trebao biti do krajnosti jednostavan svo umijeće trebalo se sastojati od onoga što danas nazivamo — softver

Glavni dio Turingove zamisli bila je beskonačno duga papirna traka podijeljena na kvadrate Mehanička glava stroja mogla se neograničeno kretati u oba smjera od kvadrata do kvadrata mogla je čitati brisati i upisivati simbole i prema simbolu kojeg je pročitala mijenjati stanje — danas bismo rekli da je mogla postaviti ili resetirati neki od svojih flag registara U skladu s postavljenim stanjem različito bi reagirala na sljedeći simbol Tu sposobnost ispitivanja podataka i odlučivanja o daljem toku na osnovi rezultata Turing je preuzeo od Babbageova zlosretna analitičkog stroja

Kad bi se u Turingov stroj uložio program za dokazivanje valjanosti nekog matematičkog teorema, stroj bi kloparao gore-dolje po traci pa ako je mogao dokazati valjanost teorema — stao bi a u protivnom radio bi dalje

Turing je na svojem konceptualnom stroju razrađivao ranije Kantove spoznaje da bi dokazao kako ne postoji konačna metoda za rješenje svih matematičkih pitanja I to je na neki način bilo dobro za matematiku jer su matematičari osjetili da će za njih još dugo biti posla Ali pravo značenje — tada još neprocijenjeno — bilo je činjenica da je postavljen okvir u kojemu će se uskoro pojaviti slika kompjutora Papirna traka predstavlja jednodimenzionalni niz memorijskih ćelija koje mogu sadržavati i program i podatke Mehanička glava je u stvari procesor a jedina je razlika iako ne fundamentalna što procesor može doći direktno do svake memorijske ćelije a ne pomicati se korak po korak duž papirne trake na kojoj se nalazi sva pamet stroja

Logičari 19 stoljeća su dokazivali da se svi matematički procesi mogu izgraditi od jednostavnih logičkih koraka Turingova zasluga je i u tome (danas je to općenito prihvaćeno) što je dokazao da stroj koji se može služiti osnovnom logikom može izvoditi i sve matematičke postupke a prema tome može imitirati i sve ostale strojeve Na tako jednostavnim osnovama koje su tako malo obećavale Turing je iako samo u mašti sagradio svemogući i univerzalni stroj

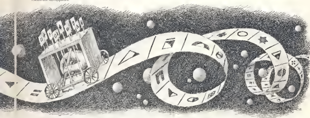
Time je stvorena misaona osnova za dalji razvoj kompjutora koja se oslanjala na dvije misli — vodilje Prva glasi *Svaka pojava u svemiru može se prikazati nizom simbola* A druga



Gore Charles Babbage (1792—1871) pokušao je sagraditi prvi mehanički računski stroj

Desno mehanički računski strojevi dostigli su krajnju granicu kompliciranosti u strojevima za diferencijalnu analizu Na sl. je stroj kojega je sagradio Vannevar Bush 1935 godine

Dolje beskrajan i nepostojeći stroj iz Turingove mašte koji nikada nije postojao ali je doveo do nastanka kompjutora



Godel i Turing pionir kompjutorstva otkrili su čudan krajolik u kojemu nema ničega osim dvaju golemih i tajnovitih monumenata. Oni što su tamo bili ne misle jednako kao ovi što su ostali kod kuće

Alan Turing (1912—1954) britanski matematičar čiji je nepostojeći računski stroj (1936) predstavljao teoretsku podlogu današnjih kompjutora. Za drugoga svjetskog rata radio je u centru savezničkih službi za probijanje njemačke vojne šifre. Počinio je samoubojstvo trovanjem, iako neki misle da je zapravo bio umoren od neke tajne službe koja je smatrala da njegova homoseksualnost predstavlja opasnost za tajne u koje je bio upućen

Svako zbivanje u svemiru može se prikazati nizom simbola. Svatko tko se danas bavi kompjutorima prihvatio je svjesno ili nesvjesno ove dvije postavke.

Turing je na Sveučilištu Princeton surađivao s Von Neumannom, a tamo je susreo i Claudea Shannona, čija knjižica 'Matematička teorija komunikacije' spada u klasična djela našega stoljeća. Shannon je otkrio nedjeljivu jedinicu informacije: bit, dva moguća stanja, DA i NE. Tako je učinjen još jedan korak prema elektronskim računskim strojevima.

Njemački inženjer Kongrad Zuse sagradio je 1937. prvi binarni računski stroj. Iste je godine Turing, neovisno od Zusea, koristio istu ideju i uz pomoć elektromehaničkih releja sagradio stroj za množenje. Ali tada izbije rat. Zuse i njegova računala rade na njemačkim raketnim projektima, a Turing postaje jedan od vrhunskih matematičara što rade u srcu jedne supertajne organizacije na razbijanju njemačke vojne šifre.

Tada, početkom rata, razbijanje šifri obavljalo se s papirom i olovkom u ruci. Ali uskoro je količina materijala postala tolika da se u britanskom centru u Bletchley Parku stalo pomišljati na automatizaciju. Turing i njegovi kolege uskoro su napravili elektromehanički stroj koji je ispitivao razne mogućnosti dešifriranja njemačkih radio signala i sredinom rata već je postojao uređaj koji je bio u stanju da zaista ispituje razne kombinacije šifri i na osnovi rezultata poduzima određene akcije. Bio je to prvi znak rađanja pravog kompjutora.

U SAD su u to vrijeme građeni slični hibridni strojevi za dešifriranje i proračune putanja artiljerijskih projektila (vidi str. 88-89), a ubrzo nakon rata napravljen je ENIAC, stroj koji je na svojih 18 000 elektronskih cijevi mogao zapamtiti 20 brojeva. Rat je veći dio svijeta ostavio u ruševinama i pobjednici su imali prečih briga nego da se bave snovima nekolicine ekscentričnih matematičara. U Britaniji u Manchesteru napravljen je stroj s mnogo nedostataka, ali i s jednom novosti: tekući podaci mogli su se na određen način pohranjivati na zaslonu katodne cijevi, što je neka vrst primitivne memorije. Elektronski top je na određenom dijelu ekrana ispisivao svijetle točke koje su predstavljale bitove. Točke bi s vremenom gubile naboj, pa ih je trebalo osvježivati prije nego što bi se sasvim izgubile.

I Amerikanci su se gotovo istodobno suočili s istim problemom, ali su njihovi rezultati u početku bili još skromniji. S obje strane oceana većina odgovornih je smatrala da nikada neće biti potrebno više od pet-šest ovakvih strojeva u čitavu svijetu.

Nakon toga je mnogo teže pratiti povijest razvoja kompjutora, jer sve manje ima dramatičnih bljeskova inspiracije, a sve više strpljiva i anonimna rada legije stručnjaka. Kompjutori postaju sve jeftiniji i masovniji proizvod i sve češće je korisnik taj koji ga usvršava i dograđuje.







Tekuci trendovi

Sadasnjost dobija svoj smisao samo ako je promatramo u svjetlu prošlosti Tu kompjutori nisu iznimka ovo neizmjerno zanimljivo područje privuklo je u posljednjih 40 godina najbolja znanstvene talente i kao što biva sa stvarima koje se naglo razvijaju i ovdje je mnogo ovisilo o slučajnostima Bilo je pokušaja da se razvoj odredi i isplanira ali kao i obično prije nego sto



bi se planovi objavili život bi skrenuo u neke druge tokove Iz toga brzog i neorganiziranog razvoja naglo su iskrsnuli neki novi koncepti uređaji međusklopovi i kompjutorski jezici Svatko tko zeli plivati u matici kompjutorskih zbivanja mora steći nekakav pojam o tim zbivanjima

Da tranzistor nije izumljen vjerojatno bi kompjutor još uvijek bio kuriozitet dostojan strahopoštovanja Ipak tranzistor je izumljen te zbog toga što je manji hladniji i pouzdaniji od radio cijevi kompjutor je postajao sve manji i uz manju cijenu — sve moćniji tako da danas možemo postaviti na svoj radni stol mocni kompjutor veličine i cijene mehaničkog pisaćeg stroja

Isto se događa i s kompjutorskim arhivima — uređajima za pohranu podataka U vrijeme nastanka ove knjige na tržištu su se uz prihvatljivu cijenu pojavili prvi Winchester tvrdi diskovi od 5 ¼ inča na koje se može pohraniti 2 do 3 milijuna riječi Leserski diskovi su još uvijek u razvoju ali izgleda da će se oni moci napuniti s nekoliko milijardi riječi i to po istoj cijeni

Proces minijaturizacije i pojeftinjenja sto traje vec dvadesetak godina najdalje je otišao u proizvodnji čipova To je tako jednoličan i siguran razvoj te je postelo gotovo zakonito da se svakih deset godina u jedan čip može smjestiti sto puta više tranzistora što opet znači da se svakih deset godina uz istu cijenu može proizvesti milijun puta snazniji kompjutor Ono sto se u pocetku cinilo kao jednostavan i dosta delek tehnologijski trend pokazalo se da ima odlucnog utjecaja na svijet Da su Babbage Godel ili Von Neumann doživjeli samo dio toga što se danas događe bili bi u najmanju ruku zbunjeni

Svaki je kompjutor opremljen svojim BASI-Com a svaki programer što je sastavljao BASIC smatrao je da ga mora poboljšati i proširiti Stoga je svaki kompjutorski displej drugačije organiziran i svaki se disk drugačije kontrolira Moglo bi se pomisliti da je jedino tastatura stan-

Gore lijevo ENIAC prvo američko računalo s elektronskim cijevima sagrađeno 1856 godine

Lijevo profesor F C Williams i Tom Kilburn za kontrolnom pločom britanskog računala Mark I što su ga zajedno konstruirali na Sveučilištu Manchester Smatra se da je 21 6 1948 godine u ovom računalu prvi put u svijetu radio jedan program na istoj osnovi kao danas Programiranje se obavljalo u 32 bitnom kodu pomoću preklopnika na prednjoj ploči

Na maloj slici gore John Von Neumann (1903—1957) uvijek elegantan mađarski izbjeglica koji je vodio početke razvoja kompjutora u Americi

Gore Claude Shannon (1916—) iz američkih Bellovih laboratorija, otkrio je 'bit' nedjeljivu informacijsku jedinicu Na slici ga vidimo pri demonstraciji prvog kompjutorom vođenog miša koji nalazi put iz labirinta

dardna ali i tu se možemo prevariti čak i najjednostavnija operacija kao što je DELETE (briši) na svakom stroju drugačije funkcionira.

Svatko zna da nakon što je nešto natipkao mora pritisnuti tipku RETURN da bi računalo uzelo u postupak ono što je otipkao. Međutim, na nekim je računalima ta tipka označena kao NEWLINE, na nekim kao ENTER, a neka računala imaju obje tipke, ali s različitim funkcijama. Sve to može izazvati zabunu, pogotovo kad se piše kakav univerzalni program koji nije namijenjen nekom određenom računalu.

Ima još jedna neobičnost koja potječe iz vremena kad nije bilo video terminala i kada se s kompjutorom komuniciralo posredstvom teleprintera. Ako bismo naime pri tipkanju napravili neku pogrešku koju je trebalo izbrisati, glava teleprintera se nije mogla vraćati na isti način kao što se vraća kursor na ekranu, već bi ponovo ispisale izbrisane znakove i zatvorila ih među kose zagrade. Na taj način se potvrđivalo brisanje. Tako se moglo dogoditi da ste svojem računalu postavili pitanje „Kako je vaš (ocat) otac?" Iako se teleprinteri već godinama ne koriste za komuniciranje s računalima, ipak još uvijek postoje jezici i operacijski sustavi koji rade na ovakav način. Još je nezgodniji ostatak prošlosti onaj koji pri editiranju obrađuje redak po redak, a moglo bi ih se nabrojiti još nekoliko jednako tako neprikladnih.

Srećom, svijet kompjutora je također podvrgnut Darwinovoj prirodnoj selekciji, jer kapitalistički sustav neumoljivo odabire najbolje i najjeftinije. Možemo se nadati da će u konkurenciji ostati zaista najbolji — ili barem najmanje loši. Konačno će valjda kao standard ostati samo jedan sustav, a mi korisnici možemo samo strpljivo čekati dok prohuje sve nepogode.

Analogije s Caxtonom

Prvi su kompjutori bili slabašni, skupi i teško su se programirali, ali kako su godine prolazile postajali su sve moćniji i jeftiniji, ali ne mnogo jednostavniji za rad.

Kompjutor nije ništa drugo nego stroj za izvođenje programa, kao što je i tiskarska preša tek sredstvo za reproduciranje autorovih misli. Velikotražne programe za osobna računala možemo usporediti s izdavačkim bestselerima namijenjenim najširem krugu čitatelja. Ranih dana kompjuterizacije programe je pisala mala izvrsno plaćena kompjutorska elita, čiji je glavni interes bio da sačuva svoje pozicije, a programe su kupovali oni koji nisu htjeli ili nisu znali sami programirati. Sve je to podsjećalo na stanje koje je vladalo u počecima tiskarstva.

Prije nego što je William Caxton* otvorio svoju tiskaru, knjiga je bila (a počelo je to još davno prije Rimljana) jedan od najskupljih predmeta koji je ljudska ruka stvorila, dostupan samo uskom krugu bogatih. Ako je netko i bio dovoljno bogat da kupi knjigu, događalo se da nije znao čitati, jer je cijeli život potratio ratujući. Ako je ipak želio saznati što piše u knjizi, morao je platiti nekom fratru da mu je čita naglas. Zar to nije slično direktoru moćne kompanije koji kupuje u paketu kompjutor i programere i kojega ne zanimaju ni kompjutori, ni programeri. On očekuje samo rezultate. (Vrlo često jedino opipljivi rezultat je u prestižu. Nedavna istraživanja su pokazala da se u posljednjih 10 godina samo u 50 % slučajeva isplatilo zamijeniti ljudski rad kompjutorom.)

Ako vam je u ono vrijeme trebala neka knjiga (kompjutorski program), valjalo je otići do obližnjeg samostana (IBM, ICL, itd.) i zamoliti dobrog opata da preporuči nekog pisca. Opat bi tada pozvao patra Jerolima (programera) iz njegove ćelije i — kad bi nakon nekoliko godina posao bio gotov, stajao bi 20 000 kruna.

Tada se Caxton uključio u posao. Njemu je za tiskanje tuceta stranica trebalo isto toliko vremena koliko Jerolimu da naoštri pero. Dok bi pater Jerolim naslikao prvi inicijal, Caxton bi otisnuo cijelu nakladu i prodao je po 50 centi primjerak (tadašnja cijena Canterburyjskih priča).

Dobri stari opat je s razlogom bio zaprepašten. I dalje je svakome tvrdio da knjiga ne može koštati manje od 20 000 kruna i da je besramna laž tvrditi suprotno. Za ime Boga, zar jedan gentleman treba znati čitati! Ako hoće nešto pročitati, on (opat) poslat će mu redovnike (uz naplatu).

Ali, dakako, nije se tako završilo, jer su plemići i njihovi potomci naposljetku naučili čitati. Opata i njegove redovnike, proizvođače tako izuzetno skupe literature, pregazilo je vrijeme. U kompjutorskom svijetu su sve do nedavno vladali slični odnosi između onih koji plaćaju, a ne znaju, i onih koji su plaćeni, a znaju. Iako se stavovi brzo mijenjaju, ipak još neko vrijeme treba živjeti s posljedicama takvog načina mišljenja. Oni što su pravili i programirali kompjutore bili su skloni da svoj posao mistificiraju, a to istini za volju i nije bilo tako teško. Kompjutori su, istina, toliko različiti od svega što smo do sada poznavali pa treba vremena da se na njih priviknemo, jer tu nema nikakvih osobitih tajanstvenosti. Postoje doduše neke zapreke, jer nije sasvim jednostavno ovladati novim medijem, ali vrijeme je da budu srušene posve umjetne barijere što su godinama pažljivo građene s jedinim ciljem da se svakom potencijalnom pridošlici onemogući ulazak u posvećeni svijet.

Nadalje, treba uzeti u obzir mnogostrukost znanja i interese korisnika kompjutora. Prije pojave mikrokompjutora bilo je u svijetu tek nekoliko tisuća računala i nekoliko desetaka tisuća ljudi koji su se njima bavili. Mala, dobro plaćena grupa stručnjaka lako je prihvaćala zajedničke standarde za jezike, komunikacijske protokole

* Caxton, William (1422–1491) vlasnik prve tiskare u Britaniji koji je 1475. tiskao prvu knjigu na engleskom jeziku. Kasnije je tiskao još oko 100 naslova.

formate diskova i da ne nabrajamo — sve što bi IBM napravio ostali su prihvaćali kao bogzna kako promišljen i savršen standard

U posljednjih pet godina stvari se radikalno mijenjaju U svijetu danas ima nekoliko milijuna mikrokompjutora i nekoliko stotina raznih proizvođača Malo je njih odgojeno na velikim računalima a mnogi od njih su "izmislili standarde koji su već postojali

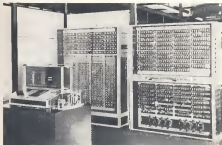
Ovo je opet nalik prvim danima tiskarstva kad nitko nije imao jasnu predodžbu o tome kako bi knjiga trebala izgledati Događalo se da naslovna strana bude na kraju knjiga a ne na početku, kako je danas uobičajeno Ovisilo je o pukom slučaju da li će pisac tiskar i izdavač biti poimence navedeni a proći će još dosta vremena dok se ove tri posve različite funkcije budu definitivno uobličile

A najgore je bilo to što nije bilo čitalačke publike Umjesto milijuna ljudi što danas kupuju knjige u Caxtonovu svijetu samo je šačica ljudi umjela čitati Zamislite kako se jedan čovjek osjećao s hrpom novih Canterburyjskih priča na polici a nigdje nikoga na vidiku koji bi to znao cijeniti

U ranim danima tiskarstva dovoljno je bilo imati prešu i biti pismen pa je tako gotovo svatko mogao napisati i izdati knjigu Otada je prošlo mnogo vremena i tiskara više nije usko grlo proizvodnje knjiga a sve je manje onih koji mogu napisati dobru knjigu a još manje onih koji je znaju urediti propagirati distribuirati i općenito pretvoriti u dobru ideju i dobru zaradu

Kompjuterizacija upravo izlazi iz Caxtonova doba Opada važnost hardvera (tiskarske preše) i pojavljuje se nova kompjutorska publika koja zna razlikovati dobar program (knjigu) od lošega Nastaje i jedna nova profesija — izdavač softvera — osoba koja odabire priprema za izdavanje oglašava i distribuira softver Sve je to sasvim drugačije od onoga što se moglo naslutiti daleke 1940 godine





PROIZVODNJA ČIPOVA



1 Tehničarka na digitalizatoru unosi ispravke u nacrt jednog prilično jednostavnog integriranog sklopa.
2 U prostorijama u kojima nema ni trunka prašine radnice pomoću mikroskopa usklađuju fotografske maske prije svake faze litografskog postupka.
3 Silicijske pogače (vafeni) se u kompjutorski upravljanim vakuumskim pećima izlažu bombardiranju (dopiranju) atomima metala koji u svojoj atomskoj strukturi imaju po jedan atom više ili manje od silicija.
4 Pladanj sa silicijskim pločicama ulaža se u peć za obradu pod visokom temperaturom. Slika jasno ilustrira temeljni prodiam proizvodnje čipova: što je više što manjih čipova na jednom pladnju, to je proizvodnja veća, a cijena manja.
5 U svakoj fazi proizvodnje čipova treba postaviti novu fotografsku masku preko strukture koja je već gotova. Na slici vidimo tehničarku za TV mikroskopom kako kontrolira položaj maske.
6 Na tankoj silicijskoj pločici odjednom se pravi nekoliko stotina komada čipova, koji se odmah pomoću kompjutora kontroliraju i nakon toga razdvajaju.
7 Mikroprocesorski čip se lijepi na donji dio plastičnog kućišta, a kontakti se pomoću tankih vodova ručno povezuju s nožicama kućišta.
8 Usprkos visokoj tehnologiji još uvijek u proizvodnji čipova ima dosta ručnoga rada. Na slici vidimo kako tehničar pod mikroskopom spaja nožice kućišta s kontaktima na čipu.

Proizvodnje čipove je neobična mješavina vrhunske tehnologije i obične kuhinje. Prije nego što čovjek uđe u takvo postrojenje, oblači poseban ogrtač koji ne ostavlja dlačice, prolazi pokraj detektora radijacije i ulazi u kondicionirane prostorije, čišće od svih u kojima je ikada bio. Ušavši mogao bi pomisliti da se našao u istočnjačkom haremu: lijepe žene pokrivena lica tako da im se vide samo oči, u nečujnoj gumenoj obući, raznose naokolo zdjele pune silicijskih pogača (vafera) i čipova, stavljaju komadiće silicija u kompjutorski upravljane difuzijske peći i rukuju mikroskopima pažljivo gledajući u TV ekrane. U ovoj kuhinji perač posuđa ima doktorat nauka, a krastavac koji upravo razrezuju je ogromni kristal čista silicija. Njegove kriške se poliraju do optičke savršenosti s tolerancijom od nekoliko mikrona.







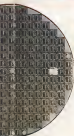
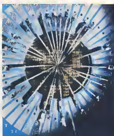
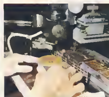


Umjesto stroja za rezenje kruha dijamantska pila reže čipove pa kada napravi poput vlasi tanke rezove, neki snažan čovjek udara batom po pločici silicija i rezdvaja čipove. Fini ženski prsti tada skupljaju i probiru buduće procesore kao da prebiru grašak.

Dugi red djevojaka za mikroskopima kao da je zaposlen pravljenjem nakita, ali nije to nakit: to čipove stavljaju u kućišta i povezuju ih s kao vlas tankim vodičima sa izvodima integriranog sklopa.

Jedna namrgođena dama poslužuje mehaničkog pauka: integrirci lete kroz njene prste i ispravni padaju u jednu, a neispravni u drugu košaru. Kontrola je vrlo stroga, jer se i ovdje kao u svakoj dobroj kuhinji živi od provjerene kvalitete.



Prva slika desno Hewlett-Packard, doajen proizvođača kompjutora 1983 je najavio 32-bitni procesor. To doslovno znači da na vaš radni stol dolazi veliki kompjutorski sustav s toliko kompjutorske snage da niko ne zna što bi s njom započeo. Usporedite pravilnost strukture Packardova procesora s prilično zamršenom strukturom 16-bitnog procesora tvrtke Texas Instruments na str. 13

Druga slika desno eksperimentalni integrirani sklop Josephsonove sklopke — super brzog uklopnog elementa koji radi na temperaturi apsolutne nitlice

Na suprotnoj strani sheme mikroprocesorskog čipa povećana oko 300 puta. Svaki sloj budućeg čipa crta se u drugoj boji i prije nego što će otići u proizvodnju sve veze moraju se pažljivo ispitati. Sa sve jačim smanjivanjem razmaka između elemenata i linija sheme postaju zamršenije, a posao oko njihove projektiranja sve teži i dugotrajniji. U tome je od velike pomoći CAD kompjutor (vidi sl. 102), kojemu treba samo reći kakva je namjena čipa, a on će odabrati najbolju kombinaciju logičkih sklopova i gotovo samostalno konstruirati čip

Treća generacija mikrokompjutora bit će snabdjevena RAM memorijom čiji će se kapacitet mjeriti megabajtima. Kad bi se sadržaj svake memorijske lokacije ispisao na bibliotečnim karticama i one se poredale jedna do druge uzduž Temze, niz bi bio dugačak nekih 100 kilometara — kao udaljenost od Londona do Oxforda





Čip u računalu pomoću kojega sam pisao ovu knjigu, a vjerojatno i u računalu kojem se vi služite, spada u 6-mikronsku klasu. To znači da su elementi i vodovi u njemu razdvojeni razmakom širokim 6 mikrona (jedan mikron iznosi milijunti dio metra), što je po prilici dvadesetina debljine papira na kojem je tiskana ova knjiga. Ova mjera je vrlo važna jer daje mnogo informacija o karakteristikama samog čipa.

Za vrijeme pisanja ove knjige na tržištu su se već pojavili čipovi izrađeni tehnologijom koja omogućava razmake od 2,5 mikrona, što znači da jedan tranzistor zauzima 40 kvadratnih mikrona, te ih na jedan čip može stati 24 000. Iako su već 6-mikronski čipovi pravo čudo tehnologije, proizvođači i dalje ulažu velike napore da tu dimenziju još više smanje. Kao što smo vidjeli na stranicama 165—166, cijena čipa malo ovisi o onome što se na njemu nalazi, pa što su razmaci između elemenata manji, to se na čip jednake veličine može smjestiti više još bržih tranzistora.

Ako se širina razmaka smanji za dvostruko, broj tranzistora koji se mogu smjestiti na istu površinu bit će četiri puta veći. Nadalje, budući da se broj elektrona u vodovima također dijeli sa četiri, oni se kreću četiri puta brže, što ukupno povećava brzinu 16 puta. Sve to dakako uvjetuje smanjivanje napona napajanja na polovicu prijašnje vrijednosti.

Konačni učinak je da za istu cijenu dobijamo čip koji radi osam puta brže, što također znači da se za osminu cijene može napraviti kompjutor istih mogućnosti. Po tako niskoj cijeni prodavat će se više primjeraka nego ranije i prema zakonu tržišta, koji kaže da se uz polovicu cijene učetvorostručuje prodaja, kompjutor postaje još jeftiniji jer se proizvodi u većim serijama.

Kao i svi drugi, tako i američko Ministarstvo obrane uviđa prednosti minijaturizacije, pa unatrag nekoliko godina potiče proizvođače da i dalje razvijaju VLSI tehnologiju (Very Large Scale Integration — integracija vrlo velikog stupnja) tako da se već danas proizvode čipovi klase 0,5 mikrona. Takav procesor bi odgovarao snazi od dvadesetak računala iz serije IBM 370. Hewlett-Packard je već najavio 1-mikronski 32-bitni procesorski čip, što znači da će se kompjutor snage velikih računskih sustava moći smjestiti na radni stol.

Slaba strana malih razmaka između elemenata je u tome što je teže proizvesti ispravan čip. Pogreška koja se tolerira kod 5-mikronskog čipa, kod 2,5 mikronskog je neprihvatljiva. Proizvođači smatraju da se proizvodnja isplati ako je 30% čipova s jednog silicijskog vafera ispravno i sad se boje što će se dogoditi ako u proizvodnji novih čipova velike gustoće samo jedan na nekoliko vafera bude ispravan.

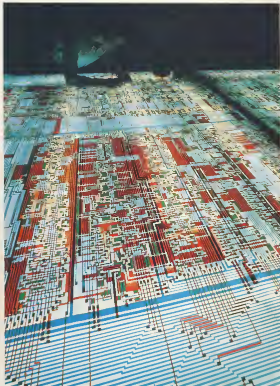
Mnogi su uzroci neispravnosti kod ovako gustih čipova. Fotografske maske se različito deformiraju, pa se kasnije ne poklapaju, a najsitnije čestice prašine, koje prolaze i kroz najgušće filtre, lijepe se na maske i upropašćuju čipove. Čak je valna dužina svjetla pod kojim se maske kopiraju predugačka, te dolazi do ogiba svjetla što razara oštrinu nezamislivo fine fotografske maske.



Granice minijaturizacije

Kao što je navedeno u početku ovoga poglavlja, ako se naprave manji razmaci između elemenata čipa, u principu se može napraviti brži kompjutor, što znači ili ista snaga za manje novca ili više snage za isti novac. Ali prije ili kasnije priroda će zaustaviti ovaj napredak, pa je zanimljivo pogledati gdje su granice minijaturizacije.

Prvi problem odnosi se na činjenicu da se čipovi zapravo prave postupkom tiskanja, neko-



liko predložaka neizmjerno fine strukture otiskuje se točno jedan preko drugog na vrlo maloj površini. Očigledno je da se to može uraditi jedino fotografskim postupkom. Maske se crtaju u velikom mjerilu, a potom fotografski smanjuju na sićušne dimenzije čipa i prenose na čip tako da se silicij prekrije foto-osvjetljivim lakom i eksponira preko maske. Eksponirani dijelovi se kemijski otvrdnjuju, a ostatak se ispire. Isprani dijelovi se nagrizaju kiselinom (jetkaju), a oni pokriveni ostaju netaknuti i predstavljaju vodove i ostale elektronske elemente čipa.

Sve dok razmaci među dijelovima nisu manji od nekoliko mikrona, proces funkcionira odlično. Nevolja počinju kad se smanje na 1 mikron, jer je tada valna dužina svjetlosti jednaka širini razmaka, pa zbog efekta ogiba (difrakcije) svjetlosti rubovi više ne mogu biti oštri, te se fina struktura maske zamućuje. Rješenje je nađeno primjenom svjetlosti kraćih valnih dužina, pa se sada koristi ultraljubičasta svjetlost valne dužine od 1/10 do 1/1000 mikrona ili rentgenske zrake valne dužine ispod 10^{-14} mikrona.

Ali što su valne dužine svjetlosti kraće, to su složeniji fotografski postupci i jedno od mogućih rješenja je potpuno napuštenje fotografije i prelazak na zrake elektrona dovoljno kratke valne dužine, koje se mogu precizno upravljati pomoću magnetskog polja. Elektronski mikroskopi dugo se već služe ovom tehnikom, pa zašto na isti način ne bi bilo moguće prenijeti strukturu čipa direktno na pločicu silicija. Jedina mana tog postupka je u tome što je jako spor. Kad pomislimo da za jedan čip treba desetak slojeva, od kojih svaki izgleda poput plana nekog velegrada, postaje jasno da je prednosti fotografskog postupka teško napustiti. Stari postupak možemo usporediti s tiskanjem poštanskih maraka, kad se odjednom otiskuje cijeli list s nekoliko stotina maraka (čipova na jednoj silicijskoj pogači), a novi bi se postupak mogao usporediti s tiskanjem svake marke posebno.

Cijeli smisao manjega čipa je u tome da bude jeftiniji, ali ako proizvodnja postane vrlo spora, postat će skuplji. Drugi dio problema je u linijama i elementima čipa koji se smanjivanjem približavaju veličini atoma materijala od kojega su napravljeni elementi. Svejedno koji metal je upotrijebljen, jer su svi atomi po prilici iste veličine, oko 10^{-10} metara. Kada se jedan vod smanji na širinu od samo nekoliko atoma, vrlo je vjerojatno da će tokom vremena jednostavno ispariti. Realna širina trajnoga voda u čipu ne bi smjela biti manja od dvadeseterostruke veličine atoma, što određuje donju granicu na 5/100 mikrona — 50 puta manje nego u suvremenih čipova. To znači, ako se ikada sagradi kompjutor s takvim čipovima, on će biti milijardu puta snažniji od današnjeg.

Ali ima još nekih drugih problema koji će se pojaviti prije no što se linije u čipu stanu isparavati. Naime, već kod 1-mikronskih vodova glavna poteškoća je u neodređenom položaju elektrona. Kao što je još 1927. Werner Karl Heisenberg ustvrdio, nije moguće u isto vrijeme izmjeriti i brzinu i položaj elektrona. Iako znamo kako se brzo kreće elektron duž nekog voda ultramikroskopskog čipa, ne možemo znati na kojem se mjestu nalazi. Možemo se samo nadati da je u nekom od vodova ili nekom tranzistoru ili u bilo kojem funkcionalnom elementu integriranog sklopa, ali po Heisenbergovu principu neodređenosti to je isključivo stvar slučaja. Možda se nalazi tamo gdje pretpostavljamo, ali on može biti i bilo gdje drugdje. Ova pojava izaziva zbrku i u najbolje projektiranim čipovima. Smatra se da slični efekti već izazivaju smetnje u 64K RAM memorijama.

Rješenje bi bilo da se u budućnosti služimo manjim i preciznijim kuglicama za našu elektronsku računaljku. Premda je prema ljudskim mjerilima elektron veoma sićušan, u usporedbi s fotonom on je prilično velik i nezgrapan. Foton nema mase, položaj mu se može preciznije odrediti i upravljati njime uz manji utrošak energije, pa bi svakako bio pogodnija kuglica za našu računaljku, koja bi se zbog toga manje grijala u radu i bila fizički manja.

U takvom kompjutoru poluvodički laseri zamijenili bi tranzistore. Na edinburškom Sveučilištu već je ranije razvijen uređaj u kojem jedna zraka svjetla može upravljati drugom, baš kao što u tranzistoru jedna struja elektrona upravlja drugom. Primjenom takvog uređaja bilo bi moguće arhitekturu današnjih kompjutora direktno prevesti na rad s fotonima.

Međutim, i uz pretpostavku da se naprave tako mali čipovi (poznavajući ingenioznost ljudske vrste, nema sumnje da će se prije ili kasnije to i dogoditi), iskrsavaju i mnoge druge teškoće. Prva i najznačajnija odnosi se na brzinu signala. Einstein je otkrio da ništa ne može putovati brže od svjetlosti, od 300000 km u sekundi. To nam se u prvi mah čini tako brzo da je suvišno o tome razmišljati, ali ne smijemo se zavaravati.

Kompjutor dobiva svoju vremensku jedinicu od vlastitoga sata: tik — procesor zahvaća jedan bit, tak — obrađuje ga, tik — prenosi ga. Da bi kompjutor ispravno radio, satni impulsi

Prema Heisenbergovu principu neodređenosti nikada ne možemo sigurno znati gdje se neki elektron nalazi i kojom se brzinom kreće. Do nedavna je to predstavljalo samo zanimljivu filozofsku spekulaciju, ali danas kad čipovi postaju sve manji i manji, postaje pravi problem, jer znači da elektron koji bi se trebao nalaziti na nekom određenom mjestu može isto tako biti negdje u susjedstvu. Taj efekt bi mogao postaviti granica daljem smanjivanju dimenzija i povećavanju brzine integriranih sklopova.

moraju stizati do svih sklopova istodobno ili barem unutar jedne desetine trajanja satnog impulsa. To znači da najveća udaljenost u kompjutoru ne smije biti veća od udaljenosti koju prelazi svjetlost za vrijeme trajanja jedne desetine impulsa.

Suvremeni mikrokompjutori rade sa satnim frekvencijama od 4 do 10 MHz, a za desetinu tog vremena (1/40 do 1/100 milijuntog dijela sekunde) svjetlost prevali udaljenost od 7,5 odnosno 18 metara. I to je maksimalna veličina koju smije dosegli kompjutor. Ako se žele postići veće brzine povećanjem frekvencije — kompjutor mora biti manji. Računalo koje radi s frekvencijom od 100 MHz ne smije biti veće od 1,5 metra. To ne bi trebalo biti neki osobiti problem kad i smanjivanje ne bi imalo svojih granica. Naime, pored ostalih poteškoća, smanjivanje donosi još jednu: na vrlo malom prostoru koncentrirano je vrlo mnogo komponenti, jer se radi već o milijunima tranzistora, koji se tako stiješnjeni sve jače griju, pa tu toplinu treba nekamo i na neki način odvoditi kako se ne bi dogodilo da nam kompjutor eksplodira nakon ukapčanja.

Elegantan izlaz mogao bi se naći u supervodljivom kompjutoru koji radi u kupki tekućeg helija. Jednim potezom bila bi riješena dva problema: prvo, budući da električna struja teče bez ikakva otpora kroz supervodič, oslobađa se malo topline, i drugo, cijeli se uređaj drži na 4 stupnja iznad apsolutne ništice, pa i ono malo suvišne topline nije teško odstraniti. Dakako, bilo bi nezgodno držati kompjutor u tekućem heliju, ali već danas su poznati supervodiči koji taj efekt pokazuju i kod znatno viših temperatura.

Dva uređaja trebale bi pomoći u konstrukciji supervodljivih kompjutora. Jedan je Josephsonova logika koja se osniva na dvjema pojavama, od kojih se prva odnosi na tuneliranje elektrona. Navikli smo, naime, materijale dijeliti na vodiče i izolatore, na primjer, bakar je vodič, a polietilen izolator. Kao što često biva u fizici, ovo vrijedi samo u makroskopskim razmjerima, jer ako se napravi dovoljno tanak sloj izolatora među vodičima, izvjestan broj elektrona će ipak »tunelirati«, proći kroz izolator. Uzrok tome je opet Heisenbergov princip neodređenosti, pa će prema tom principu neki od elektrona na izolatorskoj barijeri postati nesigurni i ponašati se kao da ne postoje. Druga posljedica je ako se tako tanak sloj izolatora hladi do temperature na kojem postaje supervodljiv, prestaje biti izolator: elektroni nesmetano prolaze kroz njega sve dok se ne izloži utjecaju magnetskog polja. Tada opet postaje izolator.

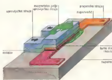
Brian Josephson je 1962. otkrio (postao je nobelovac 10 godina kasnije) da na taj način dobijamo sve elemente elektroničke sklopke. Struja koju želimo kontrolirati teče kroz Josephsonov spoj, a struja koja njome upravlja kroz zavojnicu gdje stvara magnetsko polje. Ako je upravljačka struja mala ili je nema, nema ni magnetskog polja i Josephsonova struja teče nesmetano kroz sklopku. Ako se pak tok upravljačke struje dovoljno poveća da stvori magnetsko polje, izolator opet poprima svoja izolacijska svojstva i Josephsonova sklopka se zatvara.



Centralni procesor kompjutora Cray 1, najmoćnijeg i najbržeg u svijetu. Sam procesor mora biti dovoljno malih dimenzija da bi se izbjegle pogreške uzrokovane sinkronizacijom s brzinom svjetlosti. Procesor oslobađa tako mnogo topline da ga se mora hladiti vodom. Ormari u pozadini sadrže memorije i kompjutore niže razine koji upravljaju ulazima i izlazima.

Josephsonove sklopke su vrlo brze: mijenjaju stanje u 10 do 15 pikosekundi, što bi omogućilo frekvencije do 50 milijardi ciklusa u sekundi. Naravno, kao što smo ranije vidjeli, tako brzi kompjutor mogao bi biti i vrlo malen, ne veći od 0,06 centimetara i mogao bi se sastojati od jednog jedinog čipa. Ali, na žalost, Josephsonove sklopke zasada trebaju mnogo više prostora nego tranzistori, pa se te sjajne mogućnosti još ne mogu iskoristiti. Neke veće kompanije, kao IBM, već su eksperimentirale s ovakvim uređajima, ali još ni jedne ih nije iskoristila u komercijalne svrhe.

Istraživači, opet iz IBM-a, su nedavno najavili novi uređaj — Quiteron koji radi poput tranzistora: napon na jednom izvodu upravlja strujom koja teče kroz supervodič na druga dva izvoda. Naponom upravljani sklop je poželjniji od strujom upravljanog, jer je lakše više takvih sklopova povezati u niz. Čini se također da Quiteron zauzima manje prostora, pa je pogodniji za sklopove s visokim stupnjem integracije.



Super kompjutor budućnosti (te je budućnost možda već u idućem desetljeću) hladit će se tekućim helijem. Na slici vidimo shematski prikaz Josephsonove sklopke čija se supervodljivost osniva na temperaturama oko apsolutne ništice. Strujnim tokom kroz supervodič upravlja se magnetskim poljem.

Oulleron je sličan kon vencionalnom tranzistoru s tom razlikom što se nje govo djelovanje temelji na promjeni iz supervodljivog u normalno stanje i obrat no. Nagli porast napona između S1 (plavo) i S2 (zeleno) otvara put strujnom toku od S2 na S3 (ružičasto).

Paralelno procesiranje

Postaje sve jasnije da samostalni procesori ma koliko bili mocni nisu jedino i najbolje rješenje. Čak i za tisucu puta brže i mocnije kompjutore od današnjih neki će zadaci i dalje ostati preteški.

Trebalo bi se vratiti nekoliko poglavija natrag (str 116—117) na kompjutorski vid i pogledati kako kompjutor ispituje svoje vidno polje i koliko puta uspoređuje svaki piksel sa susjednim. Najbolji suvremeni vizualni sustavi imaju možda 100000 piksela pa čak i uz uporabu najmocnijih kompjutora programi za prepoznavanje ljudi u kretanju rade satima. Jedno je to pogotovo kad napravimo usporedbu s ljudskim okom koje ima oko 3 miljuna piksela (svjetlosnih receptora) a procesira sliku za 1/25 sekunde.

Ono za čim se teži nisu samo brži procesori već više njihov broj ili točnije rečeno mogucnost paralelnog procesiranja. Uostalom na sličan način radi i naše oko: svjetlo pada na štapice i čunjice koji šalju u mozak signale o boji i intenzitetu svjetla ali obavljaju i obradu na nižoj razini — ujednačavaju smetnje, prilagođavaju se svjetlu i sjeni, zapažaju jednostavne oblike i kretanje — kao da nekoliko procesora radi istodobno.

Zamislimo kompjutor koji ima na svakoj svojoj ćeliji osjetljivoj na svjetlo, na svakom pikselu priključen po jedan procesor. Početna faza gledanja usklađivanje sa susjednim ćelijama da bi se eliminirao šum svela bi se na šest radnih ciklusa umjesto na milijune koje bi trebao samo jedan procesor.

Na sličan bi način svaki pojedinančni procesor u procesu detektiranja rubova jednostavno pitao susjedni procesor: "vidiš li istu boju i intenzitet kao ja?" Jednako i istodobno na čitavoj slici odvija se proces konstruiranja područja sličnih tonova. U času kad obrada dosegne razinu pitanja "da li je to ptica ili avion?" ista mnogostruka struktura djeluje kao banka podataka tražeci istodobno moguce identifikacije.

Nekoliko kompanija i institucija bave se problemom paralelnog procesiranja pa tako britanska kompanija GEC ima uređaj za obradu vizualnih informacija nazvan Grid chip (mrežasti čip), u kojem je međusobno povezano više pomocnih procesora, a svaki je opet povezan s dijelom vlastite memorije gdje pohranjuje podatke i program. Da bi stroj bio funkcionalan cijela memorija je dostupna glavnom procesoru koji u svaki pomocni procesor upisuje odgovarajuci dio programa: eliminiraj smetnje, odredi rubove, odredi područja istoga tona, pogledaj što imaš u svojoj bazi podataka ... itd.

Drugačija shema, razrađena na Sveučilištu Stanford, imala bi općenitiju strukturu. Procesori su raspoređeni u četiri razine. Svaki procesor na svojoj razini upravlja sa dva procesora na nižoj razini. Tako ih na zadnjoj razini ima osam. Oni kontroliraju svu memoriju i budući da se svakom od procesora može narediti da bude transparentan, to jest da samo prosljeđuje naredbe bez ikakve obrade, takva hijerarhija otvara mnogostruke mogućnosti. S jedne strane pojedini procesor najviše razine ima kontrolu nad svim procesorima niže razine i radi poput obična procesora. S druge strane, svi procesori više razine djeluju paralelno, i naravno moguća je bilo koja kombinacija između te dva ekstrema.

Dakako, problemi pri paralelnom procesiranju nisu jednostavni. Nije teško sabrati gomilu procesora i međusobno ih povezati, ali je teško

Desno: jedna moguća shema povezivanja procesora: glavni procesor (1) povezan je s vanjskim svijetom i sa dva druga procesora (2). Svaki od ta dva je povezan sa četiri nova (3), a ovi pak sa osam sljedećih (4). Ovih osam procesora najniže razine povezani su sa osam segmenata memorije koji sadrže program i podatke. Sve to može biti sadržano u jednom čipu i ovisno o konfiguraciji ponašali se ne različite načine. Ako se procesorima druge, treće i četvrte razine naredi da miruju glavni procesor je priključen na memoriju i radi kao običan procesor. Ako procesori prve, druge i treće razine miruju, onda osam procesora radi paralelno. Između tih dviju krajnosti može biti više kombinacija.

povezati ih tako da efikasno djeluju. Najčešće se događa da većina procesora ostane bez posla ili ga imaju previše, posebno s međusobnim usklađivanjem i pri podjeli poslova. Iako je danas već prilično jasno kako bi se paralelno procesiranje ostvarilo hardverski, još nije istraženo kakav bi softver trebalo napisati za paralelno upravljanje poslovima. Relativno lako je uskladiti rad nekolicine jednostavnih procesora na primjer za vizualnu analizu, međutim mnogo je složenije uskladiti istovremeno izvođenje više raznih zadataka s obradom velike količine numeričkih podataka.

Nije se daleko došlo ni sa razvojem programskih jezika za paralelno procesiranje, djelimično zato što se paralelno procesiranje simulira na kompjutorima s jednim procesorom. Tako je paralelno procesiranje za sada još dijete koje će tek tokom godina postati nosilac razvoja kompjuterizacije.

Eksplozija kombinatorike

Do sada smo razmatrali, i to na prilično diletantski način, mogućnosti povećanja moći i brzine kompjutora. Zanimljivo je baciti pogled na naličja problema i zapitati se da li postoji neki problem koji je i za najmoćniji kompjutor nerješiv? Odgovor je jednostavan — postoji. Jedan od njih je šah. Imajući na umu da su kompjutori pametni i pedantni, može se pomisliti da ih se može naučiti igrati šah tako da ih se naprosto nakljuka pravilima šaha i pusti da istraže sve mogućnosti svakoga poteza. Nakon toga kompjutor bi morao biti kadar odabrati najbolje poteze koji će dovesti do sigurne pobjede.

Međutim to nije tako lako. Svaki potez se prosječno može odigrati na tridesetak načina, a svaki od njih se grana na trideset novih u drugom potezu, pa na novih trideset u trećem itd. Računalo, dakle, mora istražiti u prvom krugu 30, u drugom 900, trećem 27000, četvrtom 300000 poteza ... to dakako ubrzo izmiče kontroli, pa je lako zaključiti zašto ovladavanje šahom ostaje izvan granica dosega kompjutora našeg vremena.

Slične poteškoće javljaju se i u mnogim drugim područjima. Zamislimo trgovačkog putnika koji treba posjetiti stotinjak gradova, pa bi htio znati kako da ih sve obiđe, a da pritom prevali najmanju kilometražu? Na stranici 145 vidjeli smo kako je teško navesti kompjutor da sâm smisli rješenje nekog problema. Jedini način na koji se može riješiti ovaj problem je da se računalo navede na izvršenje jedne akcije iz svojeg repertoara, zatim druge, treće itd., pa da se na koncu vidi da li je ili nije nađeno pravo rješenje. Ali to bi bilo isto kao kad bismo od majmuna očekivali da lupkajući po pisaćem stroju napiše klasično djelo svjetske književnosti.

Vječnost je duža od trajanja svemira. Hans J. Bremmermann, čovjek sa smislom za neporecive argumente, dotakao je problem šaha, problem trgovačkog putnika i još mnoge na sljedeći način: maksimalna brzina obrade podataka kompjutorom koji teži m grama iznosi mc^2/h, gdje je c brzina svjetlosti, a h Planckova konstanta. To iznosi oko 10^{47} bitova u sekundi po gramu. To je mnogo ako se promatra iz perspektive programa za obradu teksta, ali za igranje šaha to je premalo. Da bi omogućio blio kakvu raspravu o težini kompjutora, Bremmermann jednostavno pretpostavlja da postoji kompjutor koji ima masu cijelog svemira, težak 10^{55} grama. Svemir postoji već 20 milijardi godina, ili $6{,}3 \times 10^{17}$ sekundi. S kompjutorom navedene veličine ne bi se za to vrijeme moglo obaviti više od 10^{120} bitova. Bremmermann kaže da svaki problem čije rješenje zahtijeva obimniju obradu podataka od granice koju je postavio ovim hipotetičkim kompjutorom možemo smatrati transkomputabilnim.

Suma od 10^{120} bitova čini se priličnom, ali se ona u stvari već nakon nešto više od desetak šahovskih poteza iscrpljuje. Svemir u ulozi kompjutora, radeći od svojega prapočetka, ne bi mogao riješiti ni problem trgovačkog putnika i njegovih stotinu gradova. Izgleda da su mnogi problemi iz područja umjetne inteligencije također izvan moći kompjutora.

Možda to zvuči pesimistički, ali sve što je rečeno znači jedino da se grubom silom ništa ne rješava. Čovjek mnoge stvari (poput igranja šaha) radi odlično. Ni u šahu, a niti u drugim životnim situacijama mi ne istražujemo sve mogućnosti da bismo na koncu eliminirali one nepovoljne. Nas prema cilju često vodi neko tajanstveno svjetlo koje nam pokazuje najbolji put. A programiranje toga svjetla je izvan dosega i najsjajnijih umova.

Možda ćeš ti, cijenjeni čitatelju, pronaći odgovor i postati heroj našeg vremena.

MASOVNE MEMORIJE

Manji brži i moćniji čipovi samo su jedan dio priče jer se kočnica šire primjene osobnih računala nalazi u slaboj memorijskoj podršci suvremenih kompjutora Još prije nekoliko godina moglo se samo sanjati o memoriji od 240 kilobajta na jednom disku a danas se 10 megabajta (40 puta više) smatra potpuno normalnim Ali čak i takav kapacitet nije dovoljan u usporedbi sa stvarnim potrebama

Na policu dugačku 4 metra može se smjestiti po prilici 100 knjiga svaka oko 50000 riječi što odgovara količini oko 30 megabajta kompjutorskih podataka U ladicu standardnog uredskog ormarića može se staviti oko 50 fascikala svaki sa po 100 listova papira svaki papir sa 300 riječi teksta — ukupno oko 9 megabajta Svaki službenik će vrlo brzo ispuniti takvu ladicu pa bi za šestoricu službenika trebao kapacitet od 100 do 200 megabajta ako bi sasvim prešli na kompjutorsku obradu podataka

Zbog toga bi bilo potrebno razviti uređaj za masovnu memorijsku podršku daleko većeg kapaciteta od današnjih magnetskih diskova Jednu mogućnost nudi vertikalni magnetski zapis (vidi str 42—43) kojim bi se kapacitet memoriranja povećao 40 puta što znači da bi se na jedan Winchester disk moglo pohraniti 1 5 milijardu bajtova (1 5 gigabajt)

Druga mogućnost koja će vjerojatno biti prije realizirana je laserski disk uređaj u osnovi namijenjen bilježenju televizijskih programa s ogromnim brojem podataka Televizijski signal u boji širine pojasa oko 8 MHz teško je digitalno kodirati u manje od 8 Mb/s ili 1 MB/s To znači da jedan sat programa zauzima 3600 megabajta (3 6 gigabajta) dvostruko više od najboljih Winchester diskova

Laserski disk bilježi digitalne signale nizovima udubina spiralnoga toka kao na gramofonskoj ploči Fino fokusirana laserska zraka prelazi po tim udubinama i pošto nema nikakvog fizičkog kontakta između mehaničkih dijelova nema ni trošenja ploče Mehanički princip uređaja dao bi se vrlo lako prilagoditi potrebama kompjutora tim prije što je okretanje diska i pomicanje glave u osnovi slično standardnoj magnetskoj pohrani i čitanju kompjutorskih podataka.

Sve do nedavna mana je bila u tome što se laserski disk mogao samo čitati, a ne i zapisivati Međutim napredak će i na tom području uskoro izmijeniti situaciju Sony je već najavio laserski disk na koji se može upisivati i sa kojeg se može čitati Plastična podloga diska je presvučena slojem bizmutova telurida na koji je nanesena folije od antimonova selenida Kad laserska zraka dodirne matiranu površinu diska energija apsorbirana u donjem sloju pretvara se u toplinu koja amorfnu strukturu gornjeg sloja mijenja u kristaličnu Na tim mjestima površina diska postaje sjajna i shodno tome proces zapisivanja daje digitalne signale u obliku bljestavih točkica na neutralno sivoj podlozi koji se mogu očitavati laserskom zrakom male energije

Izvjestan nedostatak je u tome što se zapisani podaci ne mogu brisati ali ako pomislimo da takav disk može sadržavati podatke koje bi jedan daktilograf tipkao 700 godina čini mi se da ova mana nije ni tako velika Osim toga otpada opasnost od nenamjernog brisanja fajlova pa se mana može pretvoriti u prednost Međutim Matsushita je najavila sličan uređaj u kojeg se disk može izbrisati i nanovo pripremiti za snimanje

Proizvođači laserskih diskova računaju s pogreškom od jednog bita na svaku milijardu bitova Televizijska slika ima blaže zahtjeve jer se jedan pogrešni bit odražava kao smetnja u trajanju samo 1/25 sekunde što oko jedva zamjećuje Ali za kompjutorske zapise trebalo bi ugraditi mehanizam za vrlo ozbiljnu provjeru ispravnosti zapisa kako bi korisnik bio siguran da je broj pogrešaka u granicama tolerancije

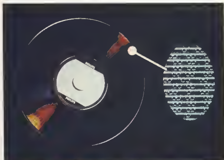
Slijedeći je problem (koji se zasada ne tiče korisnika osobnih računala) u pretraživanju tako velikih fajlova Brzina pretraživanja diska ograničena je brzinom herdverske podrške i dakako fizikalnim zakonima (vidi str 171—174) Zamislimo da se traži riječ "nosorog" na disku od čitavih 400 MB Čak kad bi to obavljao najsuvremeniji 16-bitni super-mikroprocesor još uvijek bi trebao čitav sat da se pročita cijeli disk a 8-bitnom procesoru trebao bi čitav dan da to obavi Zbog toga bi bilo potrebno iznaći mnogo savršenije metode indeksiranja pohranjenih informacija

Na prvi pogled svaka od opisanih tehnologije omogućuje pohranu daleko veće količine informacija nego što bi pojedincu ili čitavim grupama ikada trebalo Ali ne smijemo zaboraviti da bi radi sigurnosti trebalo svaki zapis nekoliko puta ponoviti a i indeksiranje upisanih podataka zauzimalo bi dosta prostora Sve u svemu proširenja memorijskog prostora što ga nude nove tehnologije i nisu tako spektakularna kao što se to čini na prvi pogled

Međutim na pomolu su nove tehnologije koje će vjerojatno konačno riješiti pitanje pohrane informacija

Jedno od mogućih rješenje koristi fundamentalni princip lasera ako se uznemiri atom pogođen svjetlom odgovarajuće valne dužine neki će se od njegovih vanjskih elektrona uzbuditi i preći u više orbite Međutim nemirni elektroni ubrzo smalaksavaju i padaju natrag na svoje mjesto ali pritom emitiraju fotone Emisija fotona može se dalje stimulirati podražajima lasera Tu drugu emisiju detektiraju fotoelektrične ćelije i čitaju bit koji jest ili nije zapisan prvotnim laserskim impulsom

Za pohranu podataka na diskove koristila bi se dva lasera manje energije, postavljena jedan nasuprot drugome pod pravim kutom Na mjestu gdje se sastaju, zrake oba lasera imale bi dovoljno energije za upis i čitanje podataka



Na desnoj strani slike vidi se povećani segment laserskog diska s mikroskopskim udubinama na površini što ih očitava tanahna laserska zraka. Na takav se disk može zapisati količina podataka ravna množini teksta od nekoliko godina neprekidna tipkanja.





Sasvim lijevo. Na ovoj simuliranoj slici (povećanje iznosi oko 10000 puta) vidi se vrh laserske zrake kako u površinu diska urezuje logičku jedinicu.

Lijevo. stotine tisuća megabajta podataka moći će se smjestiti na vrlo mali prostor kada umjesto današnjeg dvodimenzionalnog bude ostvaren trodimenzionalni zapis. Rješenje koje bi se služilo postojećom tehnologijom osniva se na zapisu podataka, bit po bit, u transparentni materijal pomoću dviju laserskih zraka što se pod pravim kutom sastaju u jednoj točki (vidi tekst).

ELEKTRONSKO SELO

"Elektronsko selo" je slikovita fraza što ju je skovao kompjutorski guru James Martin u knjizi Društvu u žici (The Wired Society). Pri tome je mislio na buduće vrijeme kada će se (i ako) jeftini kompjutorski hardver i kompjutorski komunikacijski uređaji toliko usavršiti da će povezivati ljude elektroničkim putem u jednu svjetsku zajednicu, prisnu i neformalnu poput idilična sela.

Za realizaciju plana potrebni su jeftini osobni kompjutori, sposobni za prikazivanje visokokvalitetnih slika u četiri boje, te dovoljna memorijskog kapaciteta za pohranu svih osobnih i poslovnih zapisa. Potrebne su, dakako, i komunikacijske mreže koje će jeftino i u velikim količinama prenositi podatke. Britanske pošte već postavljaju takvu mrežu na nacionalnoj razini koristeći vodiče od optičkog vlakna. U budućnosti će takve veze voditi do svake kuće i ureda, a putem satelita bit će povezane s drugim mrežama. Veliki kompjutori će nadomjestiti telefonske centrale i skupljati poruke s vrlo brzih sabirnica prespajajući ih na njihova odredišta. Dva kompjutora bilo gdje na svijetu moći će se povezati i izmjenjivati podatke, pa čak i ako se nalaze na različitim kontinentima.

Mreža neće prenositi samo tekst otipkan na tastaturi ili izvučen iz nekog fajla, veći glas, slike u boji, dijagrame, fotografije, filmove, animirane planove i nacrte. Preko iste mreže prenosit će se i televizijska slika u boji, jer je sastavni dio svakog terminala i televizijska kamera, pa će se korisnici za vrijeme komuniciranja moći i vidjeti. Oni će također imati pristup velikim kompjutorima i bankama podataka, a veći dio informacija koji je danas pohranjen u knjižnicama bit će dostupan svakom tko je uključen u mrežu.

Mrežom će se služiti svi koji moraju komunicirati s drugim osobama: direktori, poslovni ljudi i njihovi suradnici (kompjutori će tada obavljati i veći dio tajničkih poslova), arhitekti i naručitelji, trgovci i kupci, autori i izdavači, liječnici i njihovi pacijenti, generali i pukovnici. Zato će svatko moći živjeti gdje hoće i gdje mu se sviđa, a da istovremeno radi s ostalima kao da je s njima u istoj prostoriji. Nadalje, kompjutor će se ponašati kao pametan i diskretan pomoćnik kada pripremi i dobavi svaku informaciju iz vlastite ili nacionalne banke podataka.

Nema razloga da komunikacija u takvoj mreži ne bude ograničena na dvije osobe. Kompjutori će služiti kao uređaji za emitiranje — najvjerojatnije veći dio emisija bit će zabavnog karaktera (vidi str. 158—159). Budući da će mreža pomoću vlastite inteligencije moći izdvojiti svakog pojedinca, služit će i grupnim aktivnostima u školama, na sveučilištima, u klubovima, političkim grupacijama i sindikatima. Djeca će preko kućnih kompjutora pratiti nastavu u društvu s ostalim vršnjacima, od kojih su mnogi možda stotine kilometara daleko, a nastavnik čak na drugom kontinentu.

Društvene posljedice

Iako je fizičar Heisenberg izjavio "teško je predviđati, osobito budućnost", ipak smo prilično sigurni da će elektronsko selo imati jaki utjecaj na svakodnevni život u razvijenim zemljama. Na Zapadu danas više od polovice zaposlenih živi u gradovima te na ovaj ili onaj način radi na obradi informacija. U gomilama se guraju po gradskim središtima, dane provode u uredima koji su u stvari ogromni ormari za spise. Automatizacija je u tvornicama stvorila nove odnose između radnika i službenika, očigledno u korist ovih drugih, i sve će više sredstava trebati da bi službenici stizali do svojih radnih mjesta i vraćali se kući. Zato se postavlja pitanje ima li smisla ljude približavati izvorima informacija ako se informacije mogu približiti njima?

Naravno, ljudi će se i dalje družiti, raspravljati o poslovima i izmjenjivati iskustva, ali to mogu raditi i izvan klasičnih ureda. Veći dio posla obavljati će kod kuće ili blizu nje. Pošto će mnogi nerado provoditi 24 sata u vlastitoj kući, obavljat će svoj posao negdje u blizini, u zajedničkim radnim prostorijama gdje će svaki uz rad ujedno uživati u društvu s ostalim službenicima. Dakako, oni koji žele, povući će se u osamu i tamo voditi intenzivan profesionalni život. Mogu se očekivati velike posljedice i u gradovima: promet će se smanjiti, poslovni prostor također, a u selima će se stvoriti male zajednice uredskih službenika, koji će samo povremeno odlaziti u gradove da se licem u lice nađu s kolegama, ali će veći dio vremena provoditi na svojim redovitim zadacima uz kompjutor.



DO KRAJA STOLJEĆA CIJELI SVIJET ĆE BITI JEDINSTVENO ELEKTRONIČKO SELO. UDALJENOST VIŠE NIJE ZAPREKA POLITICI, RADU, UČENJU I IGRI. EVO SKICE JEDNOG OD MILIJUNA RAZGOVORA ŠTO ĆE LETJETI KROZ VODIČE OD OPTIČKOG VLAKNA I ODBIJATI SE OD SATELITA

U ITALIJI, SIGNOR FARRAZZI, KONSTRUKTOR TRKAĆIH AUTOMOBILA...

MORAM ZAOKRENUTI MATICU GENERATORA ZA 14 STUPNJEVA UNATRAG!

U REDU! POSLAT ĆU VAŠ NACRT U AUTOMATSKU TVORNICU. PROTOTIP ĆE SUTRA BITI GOTOV. PRIČEKAJTE DA VAM DAM TROŠKOVNIK

U JAPANU, GOSP. TOKUGAWA, VLASNIK AUTOMATSKE TVORNICE...

TRAJAT ĆE ŠESTNAEST SATI. TROŠKOVI VEĆI ZA DVA DOLARA

U KENIJI, JIMMY CHIP, UČENIK

TAJ JE PITAGORA SIGURNO BIO NEKI RETARDIRANI TIP

U MASSACHUSETTSU, GOSPOĐICA LINCOLN, NASTAVNICA MATEMATIKE...

USPRKOS KOMPJUTORIMA RAZMIŠLJANJE JOŠ UVIJEK NIJE IZAŠLO IZ MODE

U BOLIVIJI, GĐA SANTOS IGRA BRIDŽ SA SVOJE TRI PRIJATELJICE U NORVEŠKOJ, NOVOM ZELANDU I EGIPTU

NEMAŠ ADUTA, INGRID! KAVA ĆE TI SE SASVIM OHLADITI

OOOP

I U TEKSASU, DAISY DAYGLO VODI SATOVE TJELOVJEŽBE

NAPRIJED, MOJE DAME! MORATE OSJETITI KAKO VAM SE MIŠIĆI RASTEŽU

A ŠTO DALJE?

Već početkom industrijske revolucije moglo se predvidjeti da će se pojaviti telefon, parobrod, letjelica ili pak da će doći do svemirskih istraživanja, te je svijet prema tome bio jasno usmjeren. Isto se tako može predvidjeti kamo i dokle će nas odvesti ovaj užurbani kompjutorski razvoj.

Minijaturizacija čipova — a to znači veće mogućnosti uz manji trošak — nastavit će se sve dok ne udari u fizikalnu barijeru. Kompjutori dovoljno mali da se mogu staviti u džep imat će sposobnosti današnjih najvećih računala (poput Craya), a možda čak i veće, i u istom omjeru povećat će se i mogućnosti pohrane informacija. Komunikacije će se toliko razviti da će uz odgovarajući softver svakome biti, u bilo kojem trenutku, dostupna svaka informacija.

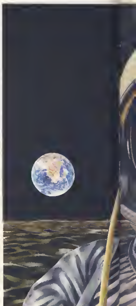
Ova tehnologija imat će neslućenu primjenu u umjetnosti. Nema razloga da se kompjutorska animacija, banka podataka, kompjutorski 'vid' i umjetna inteligencija ne udruže i ne ostvare potpuno novu umjetnost koja će film, literaturu i kompjutorske igre ujediniti u jednu cjelinu. Priče će se događati u trodimenzionalnim dekorima u prirodnoj veličini (vjerojatno pomoću holograma) s uvijek drugim licima. Razvoj priče i karaktera po volji će se mijenjati i korisnik će tako najaktivnije sudjelovati u predstavi, mijenjati njen tok, pa čak i sam nastupati.

Kad se jednom nađe u predstavi kao glavno lice ili neka sporedna uloga, kad ga zahvati matica zbivanja, sve njegove odluke i reakcije utjecat će na tok predstave i razvoj likova. Ako vam se bude učinilo da bi 'Zameo ih vjetar' bio bolji s vama u ulozi Scarlett O'Hara ili Rhetta Butlera, što vam smeta da pokušate. Ili ako su vam draži suvremeniji sadržaji, zašto da sebi ne priuštite jedno lutanje po zemlji opustošenoj atomskim ratom.

Umjesto da se samo jedan ili dvojica zabavljaju vrteći neki program na kompjutoru, u igri može sudjelovati neograničen broj sudionika. To bi moglo nadomjestiti vrhunske sportske priredbe koje bi se odvijale u arenama što postoje samo u kompjutorskoj mašti kao neke hiper-igra višekorisničkog interaktivnog sustava. Ako se ljudi ozbiljnije time pozabave, moglo bi se to pretvoriti u poslove, a i u politiku. Također nema razloga zašto kontakt s kompjutorom ne bi mogao postati taktilniji, fizički prisniji nego što je danas? Ako su kompjutori u stanju da stvore zanimljiviji i privlačniji svijet od sadašnjeg, zbog čega bi trebalo da ostanu izvan njega.

Veliki dio poslovnih transakcija kompjutori će obavljati posve automatski i samostalno. Bit će kadri da vide i zaključuju barem na razini psa-ovčara, i jedino je pitanje da li će razvoj te inteligencije pratiti i jednaki razvoj hardvera.

Primjena će svakako naći svoje mjesto i u neoružanju i ratovanju. Sasvim je moguće da će već koncem stoljeća ratovanje postati samo simboličko nadmetanje privrede i strojeva — a to može biti samo pozitivno.



Razvoj masovne strojne inteligencije i snažnih komunikacijskih sustava oslobodit će korisnike potrebe fizičkog prisustva. Ljudi će upravljati svojim poslovima i intelektualnim djelatnostima iz bilo kojeg mjesta na zemaljskoj kugli. Posljedica toga mogla bi biti da će se oni koji se ne budu uklopili u kompjutorski svijet, zbog toga što oni sami ili njihove zemlje nisu bile dorasle takvim izazovima, naći u nenadoknadivoj zaostalosti. Neminovno će nastati dva svijeta: jedan neizmjerno dobro opremljeni svijet intelektualne elite koja kontrolira informacije, politi-



ku oružje i proizvodnju, a s druge strane nekompjutorizirana, zapuštena i neuka gomila.

Nema tu zapravo novog: sve je to nalik na odnose koji su vladali prije Industrijske revolucije. Za upravljanje glupim strojevima tražilo se od svakoga neko znanje, a to je izazvalo opće prosvjećivanje i pridonijelo izjednačenju među klasama i demokratizaciji društvenih odnosa. Moguće to zvuči paradoksalno, ali moglo bi se dogoditi da nas tehnički napredak vrati u doba robovlasničkih odnosa poput onih koji su vladali dinastičkim Egiptom ili antičkim Rimom.

GDJE JE SVRŠETAK?



U svijetu se već duže vrijeme raspravlja da li će kompjutori ikada biti pametni kao ljudi pa čak i pametniji. Ima ih koji smatraju da će za stotinjak godina strojevi biti mnogo brži i pametniji od nas da će njihovo znanje biti neusporedivo veće da će raditi brže i bolje te da će biti lišeni ljubavi i mržnje onih ljudskih slabosti koje toliko koče naš napredak. Moći će sami sebe stvarati i održavati, tako da im ljudi uopće više neće ni trebati. Prvobitna ljudska inteligencija preći će u silicijska staništa i napustiti ograničenja koja joj nameće krv i meso.

Nedavno sam gledao televizijsku emisiju koju je otvorio jedan sjedokosi profesor s pitanjem da li će kompjutori kad jednom budu snagom i inteligencijom potpuno nadmašili čovjeka biti prema njemu dobronamjerni? Iz toga proizlaze još dva pitanja prvo da li kompjutor zaista može postati tako moćan ili čak moćniji od ljudskog uma i drugo ako bi ikada takav kompjutor bio sagrađen da li bi nalikovao čovjeku ili bi i dalje postojale neke esencijalne razlike?

Nije lako na to odgovoriti. Na prvo pitanje teško je odgovoriti jednostavno zato što još uvijek ne znamo kako naš mozak uistinu funkcionira. Znamo da se sastoji od stotinu milijardi neurona i da je zadaća neurona po svoj prilici obrada signala a živčani signali su zapravo brži ili sporiji nizovi impulsa. Signali se prenose od neurona do neurona preko sinapsi (spojno sredstvo između dviju živčanih stanica. Op prev.) gdje se odvijaju neki tajanstveni i nepoznati kemijski procesi. Sva u svemu čini se da neuron djeluje poput tranzistora ali ne s jednim već sa deset ili pedeset ulaza i izlaza. Unatoč desetljećima istraživanja još nitko nije u potpunosti dokučio što i kako neuron radi. Čak kada bismo i razjasnili funkcioniranje pojedinog neurona još uvijek bismo samo naslućivali na koji je način taj ogroman broj neurona u našem mozgu povezan te odakle mu tolika moć obrade podataka koja treba za govor vid ili stvaranje asocijacija.

Kad sam bio mali imao sam jednu knjigu što je prikazivala ljudsko tijelo poput tvornice, s klipovima kotlovima i upraviteljem u glavi koji kontrolira cijeli pogon. To se tada smatralo suvremenim a danas je suvremeno prikazivati ljudski mozak poput nekog kompjutora.

Smatra se da ljudski mozak može zapamtiti barem milijun pojmova ali da svatko ima još mnoga znanja što ih obično ne izražava riječima — sve ono o izgledu prijatelja o osjećaju prema obitelji o upravljanju autom o mačjem krznu o okusu vina o mirisu očeva kaputa — kada se sve to zbroji ispada da u mozgu mora biti prostora za još barem nekoliko milijardi pojmova i možda svaki neuron sadrži upravo po jednu takvu informaciju.

Čini se međutim da nema druge zapreka osim naše privrženosti navikama te se ne sagradi stroj milijun puta moćniji od svega što je danas poznato stroj koji bi radio na zasada nama nedokučiv način i koji bi ujedno bio odgo-

vor na pitanje o tome da li je moguć stroj moćniji od ljudskog mozga.

Ako se kompjutorom može oponašati svaka funkcija ljudskoga tijela, može li se onda kompjutore smatrati živim bićima? Postoji jedno gledište da sve što se ponaša poput čovjeka — i jest čovjek. Da li onda postoji neka esencijalna razlika između stroja i čovjeka?

To uopće nije novo pitanje, jer su se dugo vremena oni što su smatrali da ljudska biće nisu strojevi mogli pouzdavali u naše golemo neznanje i tvrditi da u čovjeku ima nešto izvanmehaničko. Međutim, kako se sve više produbljuju naše spoznaje o živoj i neživoj prirodi, sve je teže vjerovati da se u znanosti događa nešto protivno ili iznad prirodnih zakona. Ako ne postoji neka posebna struktura koja razlikuje ljude od strojeva, onda tu razliku treba tražiti negdje drugdje.

Neki smatraju da slobodna volja leži u slučajnostima čestica subatomskog svijeta. Možda je naš mozak samo golemo pojačalo koje pretvara život subatomskih čestica u veličine ljudskih razmjera. Taj život što ga percipiramo kao slučajno i nepredvidivo ponašanje, možda odražava zakone i događanja u nekom drugom svemiru koji je pod pravim kutom spram ovog našeg. Pa čak ako i jest tako, nema vidljiva razloga da neurone pretpostavimo tranzistorima ključnim točkama takva slučajnog ponašanja. Život se vjerojatno prije sastoji od same zamršenosti organizma, od milijardi neurona, nego od pukih megabajta RAMa kojima raspolažemo.

Postoji i mišljenje da su fizikalni zakoni samo zakoni statistike koji upravljaju događajima bez smislenosti, poput slova u ovoj knjizi, oblik kojih je slučajan i bez ikakve veze sa značenjem misli što su pomoću njih izražene.

Na koncu, koliko znamo, mozak se sastoji od običnih molekula raspoređenih na neki složeni način, koji poštuje zakone logike. Ako smo u stanju razumijevati te zakone, sve što smo do sada saznali o kompjutorima govori nam da se može sagraditi stroj koji bi se vladao po tim istim zakonima. A ako je to moguće, moguć je i živi kompjutor. Da li će to biti dobro ili zlo za nas — teško je reći, ali sigurno je da nas taj problem ne mora mučiti još koje desetljeće, ili čak stoljeće.

TABELA ASCII ZNAKOVA

	0000 0	0001 1	0010 2	0011 3	0100 4	0101 5	0110 6	0111 7	1000 8	1001 9	1010 A	1011 B	1100 C	1101 D	1110 E	1111 F
000 0	↑@ 0 NUL	↑A 1 SOH	↑B 2 STX	↑C 3 ETX	↑D 4 EOT	↑E 5 ENG	↑F 6 ACK	↑G 7 BEL	↑H 8 BS	↑I 9 HT	↑J 10 LF	↑K 11 VT	↑L 12 FF	↑M 13 CR	↑N 14 SO	↑O 15 SI
001 1	↑P 16 DLE	↑Q 17 DC1	↑R 18 DC2	↑S 19 DC3	↑T 20 DC4	↑U 21 NAK	↑V 22 SYN	↑W 23 ETB	↑X 24 CAN	↑Y 25 EM	↑Z 26 SUB	↑[27 ESC	↑\ 28 FS	↑] 29 GS	↑↑ 30 RS	↑← 31 VS
010 2	32 SP	33 !	34 "	35 #/£	36 $	37 %	38 &	39 '	40 (	41)	42 *	43 +	44 ,	45 –	46 .	47 /
011 3	48 Ø	49 1	50 2	51 3	52 4	53 5	54 6	55 7	56 8	57 9	58 :	59 ;	60 <	61 =	62 >	63 ?
100 4	64 @	65 A	66 B	67 C	68 D	69 E	70 F	71 G	72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
101 5	80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W	88 X	89 Y	90 Z	91 [	92 \	93]	94 ↑	95 ←
110 6	96	97 a	98 b	99 c	100 d	101 e	102 f	103 g	104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o
111 7	112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w	120 x	121 y	122 z	123 {	124 |	125 }	126 ~	127 DEL

KAZALO IMENA I POJMOVA S TUMAČEM

U zagradama je naveden odgovarajući engleski termin

ADRESA (adress) 18
dio naredbe kojim se određuje memorijska lokacija ili područje memorije
ADRESNA LINIJA (adress line) 25
posebni vodiči za adrese kod mikroprocesora druge generacije koji ubrzavaju rad
AKUMULATOR (accumulator) 80
dio aritmetičke jedinice procesora sastoji se od posebnog registra i pripadajućih krugova u kojima se akumuliraju rezultati aritmetičkih operacija
ALGOL (ALGOrithmic Language) 58 59 77 78
viši programski jezik za matematičke i znanstvene probleme
ANALOGNO-DIGITALNI KONVERTOR (analog to digital converter) 13 108 116 124
elektronski sklop koji analogne veličine pretvara u digitalne
ANDROID 142—145
čovjekoliki stroj robot
ANIMACIJA (animation) 33 112—115
oživljavanje crteža kao kod crtanog filma
ARHITEKTURA ČIPA 18
opće konstrukcija mikroprocesora u hardverskom i softverskom smislu
ASCII (American Standard Code for Information Interchange) 12 13 27 85 185
grupe kodnih kombinacija kojima su kodirani brojevi slova i ostali znakovi
ASEMBLER (assembler) 78 80
strojni programski jezik služi za prevođenje strojnog kôda u binarni oblik
AUTOMATIZACIJA 140 141

BABBAGE CHARLES 163
(1792—1871) Britanski matematičar i konstruktor jednog od prvih mehaničkih računskih strojeva
BAJT (byte) 12 25
grupa od osam binarnih znamenki koje se tretiraju kao cjelina
BASIC (Beginner's All purpose Symbolic Instruction Code) 12 24 58
viši programski jezik s kojim je opremljena većina osobnih računala
BAZA PODATAKA (data base) 39 49 94—97 148 156—159
masa memoriranih informacija strukturiranih tako da predstavljaju jednu specijaliziranu logičku cjelinu Izvlačenje svakog pojedinog podatka ne remeti raspored ni sadržaj baze
BINARNI SUSTAV (binary notation) 12
brojevni sustav koji raspoznaje samo dvije znamenke 1 i 0 ili dva stabilna stanja uključeno i isključeno pa je stoga najprikladniji za elektronička računala
BIT (BInary digiT) 12 25
binarna znamenka 1 ili 0 H (high) ili L (low)
BOJA 110 111
BUS (bus) 14 22 23 154 178
sabirnica skup vodiča koji prenose podatke unutar kompjutorskog sustava

C 58 77 79
viši programski jezik sličan Pascalu
CAD (Computer Aided Design) 102—105
kompjutorski sustav posebno razvijen za izradu nacrta perspektive i realne animacije
CAXTON WILLIAM 166
(1422—1491) engleski tiskar koji je 1476 tiskao prvu knjigu na britanskom tlu Izdao još stotinjak naslova
COBOL (COmmon Business Oriented Language) 58 78
viši programski jezik razvijen za komercijalnu namjenu
CP/M (Control Program for Microcomputers) 45 148
jedan od standardiziranih operativnih sustava program koji nadzire rad drugih programa

ČIP (chip) 14 15 16 19 168—170
pločica poluvodičkog materijala s integriranim krugom bez kućišta i nožica. Općenito sinonim za integrirani krug

DAISYWHEEL 10 36 37
tratinčica na engleskom Nosač slova lepezasta oblika ugrađen u novu generaciju pisaćih strojeva i printera Kod nas se koji put upotrebljava i naziv zvjezdica
DELETE 26
engl brisati poništiti Oznaka na tipki tastature kojom se poništava napisani znak
DIGITALIZACIJA (digitalisation) 34 108 151
postupak za pretvaranje analognih veličina u digitalne kao na primjer kod ploče za digitalizaciju pomoću koje se može crtati na ekranu
DIGITALNI (digital) 10
sustav kojim se podaci i veličine prikazuju pomoću brojeva ili slova
DISK (disk) 10 12 16 40—43 165 176
okrugla ploča od magnetskog materijala služi za pohranu podataka
DISPLEJ (display) 31
pokazivač indikator često i sinonim za ekran
DMA (Direct Memory Access) 42
direktni pristup memoriji procesorski čip posebne namjene koji prenosi podatke od računala prema nekom vanjskom uređaju i obrnuto mimo programske kontrole
DOPIRANJE (doping) 20
postupak u proizvodnji poluvodiča bombardiranje čistog silicija atomima joda (ili nekog drugog dopanta) što ima za posljedicu da silicij postaje pod izvjesnim uvjetima vodljiv a pod drugim ne Odatle i naziv poluvodič
DREJN (drain) 20 24 168 169
ponor odvodna elektroda kod FET tranzistora

EDITIRANJE (editing edit) 46 47 98 186
montaža mijenjanje rasporeda programskih naredbi i podataka pri programiranju
EKRAN (screen) 30 31
prednja površina katodne cijevi na kojoj se prikazuju kompjutorski podaci
ENIAC (Electronic Numerical Integrator and Calculator) 164
elektroničko računalo opće namjene razvijeno u Sjedinjenim Državama (John Mauchly i J Presper Eckert) započelo radom 1946 Sastojalo se od 18 000 elektronskih cijevi Smatra se da je time započelo suvremeno doba elektroničkih računala
ENTER 26 166
engl ući unijeti Programska naredba BASICA koja kaže računalu da je završeno unošenje podataka i da može početi obradu
EPROM (Erasable Programmable Read Only Memory) 25
vrsta memorijskog čipa čiji se sadržaj može brisati najčešće pomoću jakog ultraljubičastog svjetla
ESCAPE 26
engl pobjeći ući oznaka koja računalu kaže da ono što slijedi pripada nekom drukčijem postupku od onoga koji je do sada bio u toku
ESCHER MAURITS CORNELIS 114
(1898—1972) Nizozemac živi i djeluje u Italiji Švicarskoj i Belgiji Grafičar fantastične imaginacije s grafičkim rješenjima koja u mnogim elementima nalikuju kompjutorskoj grafici

FAJL (file) 44—47
organizirani niz ili slijed podataka
FET (Field Effect Tranzistor) 20
tranzistor s efektom polja upotrebljava se u logičkim sklopovima
FLAG (flag) 80 82
zastavica dodatni dio informacije za signaliziranje nekog stanja u programu
FLIP-FLOP 24
elektronski sklop s dva stabilna stanja nula ili 1
FLOPI DISK (floppy disk) 14 41
mehanički uređaj za zapisivanje i čitanje podataka na magnetskom disku
FORTRAN (FORmula TRANslation) 58 77 78
viši programski jezik za matematičku i znanstvenu primjenu
FRAKTAL (fractal) 80 81

GEJT (gate) 20 21 24
vrata upravljačka elektroda FET tranzistora

GENERATOR TAKTNE FREKVENCIJE (CLOCK) 14
elektronski sklop koji pomoću električkih impulsa određene frekvencije usklađuje rad pojedinih sklopova kompjutora

GIGABAJT (gigabyte) 176
jedna milijarda bajtova

GÖDEL KURT 162 164
(1906— ?) američki logičar i matematičar Rođen u Brnu u tadašnjoj Austrougarskoj docent na Bečkom sveučilištu do 1933 kada emigrira u Ameriku i nastavlja rad na sveučilištu Princeton

GRAFIČKI TABLET (graphichs tablet) 28 29 110
elektronska crtaća daska naprava koja signale dobijene od elektronskog pisala pretvara u x/y koordinate na ekranu

GRAFIKA (graphics) 32 110 112 114
crtež izrađen bilo kojim sredstvom (vođenjem kursora ili pomoću grafičkog tableta) na ekranu kompjutora

GRANANJE PROGRAMA SKOK (branching jump) 24 56 57
napuštanje osnovnog toka programa i skok na neku novu programsku liniju Ovi skokovi mogu biti uvjetni i bezuvjetni

HARDVER (hardware) 10 170—175
skup svih fizičkih (električkih i mehaničkih) dijelova od kojih su sastavljeni kompjutor i njegove periferije

HEISENBERG WERNER KARL 172 173
(1901—) njemački fizičar jedan od osnivača kvantne mehanike Nobelova nagrada 1932

HEKSADECIMALNI SUSTAV (hexadecimal notation) 12
brojevni sustav čija je osnova broj 16 Znamenke su 0 1 2 3 4 5 6 7 8 9 A B C D E F

I (AND) 20 21
oznaka za I logičku operaciju

IBM (International Business Machine Corporation) 10
najmoćnija svjetska elektronska korporacija koju je oko 1897 osnovao Herman Hollerith (prvo kao Tabulating Machine Company) izumitelj prvog mehaničkog stroja za sortiranje i obradu podataka o popisu stanovništva u Sjedinjenim Državama Danas najznačajniji proizvođač kompjutora

ILI (OR) 20 21
oznaka za ILI logičku operaciju

INTEGRIRANI SKLOP INTEGRIRAC (integrated circuity) 14—19 168
elektronska komponenta sastavljena od više elektronskih sklopova izvedenih na istoj pločici od poluvodičkog materijala

INTERFEJS (interface) 42 148
međusklop naprava za povezivanje dvaju uređaja koji se inače ne mogu direktno priključiti

INTERPRETER 79
program čija je zadaća izvršavanje korisničkog programa

INVERTOR (inverter) 20
logički sklop koji izvodi operaciju negacije (NOT)

IZLAZ (output) 12 13

JOSEPHSONOV SPOJ 173

KATODNA CIJEV (CRT Cathode Ray Tube) 30

KNJIGOVODSTVO 100 101

KOMPAJLER (compiler) 78
kompilatorski program za prevođenje naredbi u strojni kôd

KOMPJUTERIZACIJA 10

KOMPJUTORSKE IGRE 50 51

KOMPJUTORSKI JEZICI (language) 17
programski jezici sastavljeni prema određenim pravilima sa strogo specificiranom sintaksom služe sporazumijevanju između korisnika i računala

KOMUNIKACIJSKE MREŽE 146

KONTROLA JEDNAKOSTI (parity checking) 12
provjera točnosti prijenosa podataka izvan računala

KURSOR (cursor) 34 98
oznaka, obično kvadratična oblika, veličine standardnoga slovčanog znaka kojom je označeno mjesto trenutne pozicije pisanja na ekranu

LASER (Light Amplification by Stimulated Emission of Radiation) 36 173
kontinuirana emisija koherentne monokromatske svjetlosne zrake u vidljivom ili blizu vidljiva dijela spektra.

LCD (Liquid Crystal Display) 31
displej od tekućeg kristala

LED (Light Emitting Diode) 31
svjetleća dioda.

LISP (LISt Processing language) 77 79
viši programski jezik novije generacije

LOGIČKE FUNKCIJE 14 18 20 21

LOGIČKI SKLOPOVI (gates) 20

MAGNETSKA MEMORIJA (magnetic memory) 40 41
trajna memorija koja koristi osobine magnetizma za bilježenje informacija

MAGNETSKI MJEHURIĆI (magnetic bubbles) 43
vrsta magnetske memorije bez ikakvih pokretnih dijelova

MARK I 14
prvo IBM elektroničko računalo dugačko 17 i visoko 2,5 metra konstruirao ga je Howard H Aiken a pušteno je u rad 1943 na Harvardskom sveučilištu

MEGABAJT (megabyte) 176
milijun bajtova

MEMORIJA (memory) 10 12 16 17 24 25 40 43 176

MEMORIJSKA ĆELIJA (memory cell) 13 25
vidi memorijska lokacija

MEMORIJSKA LOKACIJA (memory location) 18 25
prostor u memoriji koji se može slobodno adresirati

MINIJATURIZACIJA 165

MIKROKOMPJUTOR (microcomputer) 10 11
grubom podjelom kompjutori se dijele na velike sisteme (main frame) minikompjutore i mikrokompjutore U ovu posljednju grupu spadaju sva suvremena osobna računala i poslovni kompjutori koji ne zahtijevaju specijalno obrazovanu poslugu

MINIKOMPJUTOR (minicomputer) 10
prva generacija minikompjutora pojavila se oko 1970 kao posljedica minijaturizacije kompjutorske tehnologije ali to su još uvijek veliki sistemi ograničeni na rad u velikim privrednim organizacijama i na sveučilištima Njima mogu upravljati samo visoko obrazovani stručnjaci

MIŠ (mouse) 10 28 29 46
mehaničko-elektronska naprava za pomicanje kursora po ekranu Kursor se pomiče u istim smjerovima u kojima se kotrlja miš po plohi stola

MODELIRANJE (modelling) 50 88 89
kompjutorski programi kojima se pokušava predvidjeti neke buduće događaje i razvoj situacije u ratu u nekom području znanosti u fizikalnom ili kemijskom procesu

MONITOR 24 25
isto što i displej

MOS FET (Metal-Oxide Semiconductor Field Effect Transistor) 20
poluvodički elementi odlikuju se malom potrošnjom energije i velikom gustoćom pakiranja Veći dio suvremenih čipova izrađen je u toj tehnologiji

NAPAJANJE STRUJOM (power supply) 14
uređaj koji snabdijeva kompjutor potrebnim naponima

NAREDBA (command) 58
programske naredbe strogo specificirane pravilima kompjutorskih jezika koja kompjutoru kaže što treba činiti

NEUMANN VON JOHN 77 165
(1903—1957) matematičar madžarskog podrijetla posebno zaslužan za razvoj kompjutorskih jezika

NILI (NOR) 20 21
logička operacija kod koje je uvijek na izlazu jedinica ako su sve ulazne varijable nula U svim ostalim kombinacijama rezultat je nula

OBRADA PODATAKA (processing) 10

OBRADA SLIKE (image processing) 108 109
proces popravljanja i dorade slike pomoću kompjutora.

OBRADA TEKSTA (word processing) 48 98 99
skupina posebnih programa čiji je zadatak olakšati pisanje montažu i slaganje teksta

OPERATIVNI SUSTAV (operating system) 24 44—47
procedure koja kontrolira sva sredstva unutar jednog sustava za obradu podataka. Program koji upravlja radom drugih programa

OPTIČKA VLAKNA (fibre optics) 22 154 156 157 178
fleksibilni vodiči svjetla s vrlo malim gubicima načinjeni od vlakana visokog refraktivnog indeksa (obično pairex)

PALICA (joystick) 13 28
palica pokretljiva u vertikalnom i horizontalnom smjeru za pokretanje dvaju promjenljivih otpornika. Njome se vodi kursor po ekranu

PARALELNI PRIJENOS PODATAKA 22

PARALELNO PROCESIRANJE 174

PARITETNI BIT (parity bit) 12 28
vidi kontrole jednakosti

PASCAL 58 77 79
viši programski jezik novije generacije u kojemu se nastojalo izbjeći mane starijih programskih jezika. U Pascalu se najprije izvorne naredbe prevode na međujezik ili pseudo-kôd koji se zatim prevodi u strojni oblik instrukcije za određeno mikroračunalo

PERIFERIJA (peripheral) 22 138
naziv uređaja koji rade pod nadzorom kompjutora (printer, ploter itd.)

PETLJA (loop) 56
grupa programskih naredbi koje se ciklički ponavljaju sve dok ne bude zadovoljen postavljeni uvjet

PIKSEL (pixel) 31 108
kovanica od picture cell — ćelija slike, točkica. Najmanja jedinica na koju se može razložiti elektronska slika

PL/1 (Programming language 1) 78
viši programski jezik koji kombinira prednosti COBOLA i ALGOLA

PLIVAJUĆI ZAREZ (floating point) 82
pomična decimalna točka (kod kompjutora se radi o točki a ne o zarezu), format kod kojega nije čvrsto definiran broj cjelobrojnih i razlomačkih mjesta, već se formira ovisno o rezultatu operacije

PLOTER (plotter) 10 36 39
periferija koja pod nadzorom kompjutora obavlja složene crtačke zadatke. Crta perima (ako zatreba i u više boje) koja su pričvršćena na pomične x/y saonice

POINTER 81
indirektni pokazivač adrese, odnosno pokazivačka adresa, sadržaj koje pokazuje gdje se nalazi operand

POLUVODIČ (semiconductor) 20 156
električki vodič čija je vodljivost između metala i izolatora. Među poluvodiče spada germanij, silicij, selen, bakreni oksid itd.

POSLOVNI SOFTVER (business software) 94 148—151
vrsta softvera koje se upotrebljava u uredskom poslovanju

PORTA (port) 22 28 29
naziv za kompjutorske priključke preko kojih računalo komunicira s vanjskim svijetom. Među standardne porte spada RS 232, Centronix ili IEE 488, V 24 itd.

POTPROGRAM (subroutine) 58
samostalni dijelovi programa u koje se može ući s bilo kojeg mjesta glavnog programa gdje su specificirani parametri što će ih potprogram obraditi

PRINTER 10 36 38
pisač, štampač, periferija koja pod nadzorom kompjutora na papiru ispisuje zadani sadržaj

PRISTUP (access) 40 41
pristup, doseg do neke memorijske lokacije ili magnetskog zapisa u vanjskoj memoriji

PROCESOR (processor) 14 16 46
općeniti naziv za napravu sposobnu za obradu podataka. Sinonim za centralni procesor

PROGRAM (routine, program) 16
slijed naredbi po kojima kompjutor rješava određeni zadatak

PROGRAMIRANJE (programming) 52 54
proces sastavljanja slijeda naredbi po kojima će kompjutor rješavati određeni zadatak

PROGRAMSKI JEZIK (program language) 62
jezik podjednako razumljiv korisniku i kompjutoru na kojem se piše program. Programski jezici se dijele na niže i više programske jezike

PROGRAMSKE LISTE (listing) 82
ispisivanje (na ekranu ili printeru) svega što sadrži jedan program

PROGRAMSKI PAKET (package) 17 88
program namijenjen rješavanju nekog problema napisan tako da ga mogu koristiti svi korisnici bez obzira na njihove specifične podatke ili organizaciju

PROM (Programmable Read Only Memory) 24 25
programabilna trajna memorija, memorija u koju se jedanput može nešto upisati što se kasnije više ne može izbrisati

RAM (Random Access Memory) 14 16 24 25
memorija s izravnim pristupom u koju se može upisivati i iz koje se može čitati bez ograničenja

REGISTAR (register) 16 17
spremnik kratkotrajne memorije za privremenu pohranu nekog podatka. Kompjutor može imati više registre od kojih svaki ima svoju specifičnu funkciju

RELEJ (relay) 20
elektromehanička sklopka koja pomoću male struje može indirektno ukopčati ili iskopčati veliko trošilo

RETURN 26 166
vrati se, naredba BASICa koja vraća program iz nekog potprograma na glavni programski tok. Uvijek ide u paru s naredbom GOSUB (idi dolje)

ROBOT 134 136 138 140 141
naziv iz utopističke drame Karela Čapeka R.U.R., čovjek-automat, čovjek-stroj. U suvremenom značenju svaka naprava koja može obavljati neke poslove predviđene za ljude

ROM (Read Only Memory) 12 14 24 25
memorija iz koje se može samo čitati, čiji sadržaj nije moguće mijenjati niti ga izbrisati, stalna memorija

RUKOVANJE (handshaking) 28
standardizirani hardversko softverski postupak za komuniciranje računala s vanjskim jedinicama

SEKTOR (sector) 41
dio traga na magnetskom disku ili vrpci na kojem je zapisana neka programska cjelina ili skupina podataka

SENZOR (sensor) 124—126
osjetilo, ticalo, optičko, akustičko ili taktilno koje može poslati podatke o svojem osjetu nekoj jedinici za obradu

SERIJSKI PRIJENOS PODATAKA (serial data transmission) 22

SERVO PETLJA (servo loop) 130—131

SILICIJ (silicon) 20
(Si) nemetalni prirodni poluvodički element, jedan od najvažnijih materijala za proizvodnju tranzistora i čipove

SILICIJSKA POGAČA (wafer) 168
okrugla pločica silicija na kojoj se istovremeno izrađuje više tranzistora ili čipova

SIMULACIJA (simulation) 50 102 106—107
programski postupak kojim se simulira neko događanje sa svim repercusijama kao da se zbivanje odvija u stvarnosti

SINCLAIR 14
britanska tvrtka slavna po mnogim tehničkim i tehnološkim rješenjima koja su često ispred svoga vremena. Proizvođač prvih zaista jeftinih mikrokompjutora ZX80, ZX81, Spectrum i QL. Utemeljitelj i glavni konstruktor Clive Sinclair

SKAKAČ (hopper) 132

SOFTVER (software) 10 48 49 81 94
skup programa koji kompjutoru nalažu što treba da radi

SORS (source) 20 24
izvor, vrsta dovodna elektroda FET tranzistora

SORTIRANJE (sorting) 84 85

SPECTRUM 22
mikrokompjutor niske cijene i velikih mogućnosti britanske tvrtke Sinclair

SSSD (Single Sided Single Density) 42
standardni jednostrani disk, jednostruke gustoće dimenzije od 8 inča

STEK (stack) 24 82
složaj, dio memorije u koji se pohranjuju adrese na koje

se program mora vratiti nakon neke potprogramske operacije

STRING 44 83
svaki niz podataka organiziran u sekvencu po nekom određenom redu ili ključu. Niz brojeva ili slova

STROJNI KOD (machine code) 78 80—85
direktni jezik najniže programske razine. Sav repertoar naredbi računala u binarnom obliku koji procesor izravno prihvaća

STRUČNI SUSTAV (expert system) 80 85

STRUKTURIRANO PROGRAMIRANJE (structured programming) 77

TABLA ISTINITOSTI (truth table) 20 21
tablica stanja, tabelarni prikaz logičkih funkcija

TASTATURA (keyboard) 10 14 26—27

TEKUĆI KRISTAL 31
vidi LCD

TELEKOMUNIKACIJE 158

TERMINAL 49
uređaj za dvosmjernu komunikaciju svako radno mjesto na kojem korisnik može predavati i primati podatke te komunicirati s računalom

TISKANA PLOČICA (printed circuit) 14—15 22
pločica od izolirajućeg materijala na kojoj se nalaze vodovi što spajaju pojedine elektronske komponente. Izrađuje se tehnikom sličnom tiskarskoj

TRANZISTOR 14 18 20 24

TURING ALAN 14 162
(1912—1954) britanski matematičar za drugog svjetskog rata uspješno radio na razbijanju njemačke vojne šifre. Njegovi kasniji radovi na teoretskim osnovama tzv. konceptualnog računskog stroja predstavljaju osnovu današnjeg kompjutora

TVRDI DISK (hard disk) 42 43
disk sa čvrstom podlogom ogromnog kapaciteta memoriranja

ULAZ (input) 13 21 24 44 122
prihvat podataka iz vanjskog svijeta, jedinica koja prihvaća podatke i predaje ih centralnoj jedinici

VARIJABLA (variable) 58
promjenljiva veličina, promjenljiva vrijednost (u BASICu slovo ili kombinacija od nekoliko slova abecede i brojeva od 0 do 9) kojoj kompjutor pridružuje numeričke vrijednosti što ih obrađuje

VDU (Visual Display Unit) 166
vidi display

VELIKI KOMPJUTORSKI SUSTAV (main frame) 10
vidi mikrokompjutor

VERTEKS (vertex) 118—119

VIŠI PROGRAMSKI JEZIK (high level language) 17
programski jezik kod kojega svakoj naredbi odgovara nekoliko naredbi strojnog programa

XILI (XOR) 20 21
isključivo ILI logička operacija kod koje će rezultat biti nula ako su sve ulazne varijable nula ili jedan. U svim ostalim kombinacijama rezultat je jedinica

ZASTAVICA (flag) 16
vidi flag

ZIPFOV ZAKON 67

ZUSE KONRAD 164
(1910 — ?) Berlinski inženjer konstruktor prvog (1935.) elektromehaničkog računskog stroja koji je radio na binarnom principu pomoću 2000 releja. Kasnije prelazi na elektronske cijevi i za vrijeme rata se njegovi strojevi koriste za izračunavanje podrhtavanja avionskih krila. Njemačka vojna mašinerija nije pokazivala previše zanimanja za Zuseov rad

Napomena uredništva

Dok pripremamo jugoslovenska izdanja ove knjige i našu zemlju pomalo počinju zapljuskivati valovi mikrokompjutorske revolucije Znamo da su to krupne riječi, ali zaista se radi o svojevrsnoj revoluciji Promjene koje mikrokompjutori unose u naš poslovni i privatni život neće biti ništa manje od onih, što su ih svojevremeno donijeli pronalasci parnog stroja i električne struje Već kroz nekoliko godina i u našoj zemlji kompjutori će zamjenjivati čovjeka ne samo u knjigovodstvu već i u dijelovima proizvodnje Ako se to ne dogodi zaostajat ćemo za općim razvojem

U tim okolnostima ocijenili smo da pored strogo stručnih knjiga kojih već imamo u priličnom broju i u nas postoji potreba za knjigom, koja je istovremeno stručno zasnovana a ujedno i dovoljno popularno pisana, da može poslužiti kao prvi opći vodič i priručnik za svijet kompjutora Odlučili smo se za prijevod knjige »Kompjutor u kući« Petera Launea koja je prije godinu dana objavljena na engleskom jeziku i već se afirmirala kao dosad najpopularnije pisani stručni vodič ili obrnuto — kao najstručnije pisani popularni vodič o kompjutorima

Na početku ove knjige autor je obećao da će nam otvoriti prozore s pogledom u čudesne vrtove novog svijeta što ga tvore kompjutori Mislimo da je u tome uspio pod pretpostavkom da je i čitalac sa svoje strane uložio određeni interes i trud da slijedi autora

Prilikom prevođenja na hrvatsko-srpski poštivali smo autorov originalni tekst pa su tako preneseni i neki odlomci koji više odgovaraju za anglo-američkog čitaoca nego li za našeg Ali zato smo dodali programe prilagođene za »Spectrum«, mikrokompjutor koji je daleko najrašireniji u Jugoslaviji Također smo uz kazalo imena i pojmova dodali tumačenja koja praktički predstavljaju mini-rječnik kompjutorske terminologije

Upravo s tom terminologijom imali smo najviše dilema jer je u nas ona još u povojima Preuzeti već poznatu englesku riječ u našoj transkripciji i deklinaciji ili maksimalno nastojati na pronalaženju domaće terminologije? Najčešće smo se odlučivali za termin koji je najekonomičniji i koji je već odomaćen u praksi Sam život potvrdit će ili korigirati tu terminologiju i to će tek biti konačan sud Slično kao u nogometu gol korner i aut teško će ikad ustupiti pred »zgoditkom« »udarcem s ugla« ili »ubacivanjem sa strane«, ali zato jedanaesterac više nikad neće biti »elver« niti će lijevo krilo postati »outside left« Naravno s napretkom kompjutorizacije u nas jačat će i domaća kompjutorska terminologija kojoj je i prijevod ove knjige nastojao dati svoju potporu kadgod to nije bilo na uštrb preciznosti i razumijevanja

Uredništvo hrvatsko-srpskog izdanja

Prije svega nekoliko godina kompjutori su bili tajanstvenim velom obavijeni strojevi, financijski dostupni samo bogatim i velikim poduzećima i ustanovama, a intelektualno dostupni samo visokoobrazovanim matematičarima. Sve se promijenilo pojavom mikrokompjutora. Ovi mali strojevi, gotovo džepnog formata, sposobni su obaviti radnja za koje nam je prije trebao golemi kompjutor koji je ispunjavao cijelu sobu. Istovremeno, i cijena je postala sasvim dostupna: taj moćni patuljak košta manje od malo boljeg bicikla. Milijuni obitelji u svijetu i deseci tisuća kod nas u Jugoslaviji danas već posjeduju svoj kućni kompjutor, a taj krug sve brže se širi. Tako sad i djeca u svojim domovima počinju sticati prva iskustva s ovim fascinantnim strojevima, koji proširuju moć ljudske inteligencije, štede vrijeme i fizički trud, i predstavljaju najčarobniju igračku koja je ikad izmišljena.

O kompjutorima je objavljeno obilje izvrsnih visokostručnih knjiga, ali »Kompjutor u kući« Petera Lauriea prva je uopće, koja jasnim i jednostavnim jezikom svestrano dočarava sve revolucionarne implikacije što ih eksplozija kompjutora unosi u naš život. Te je knjiga razumljiv vodič, i ujedno pouzdan priručnik. Ona nas brzo i neposredno uvodi u svijet kompjutora, objašnjava nam kako oni rade i kako ih možemo upotrijebiti, i najavljuje nam kakvu nam blisku budućnost već sada stvaraju.

NAJTRAŽENIJA
ENGLESKA KNJIGA
O KOMPJUTORIMA
SADA I NA NAŠEM JEZIKU!



Iz recenzija o engleskom izdanju ove knjige.

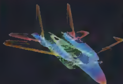
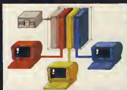
»Autor je ovu knjigu namijenio onima koji su upravo kupili ili žele kupiti kućni kompjutor, ali po mojem mišljenju to je knjiga za svakoga kojeg zanima današnji svijet i njegova sutrašnjica. Peter Laurie je otvorio prozore u niz kompjutorskih vrtova toliko uspješno, da će to sigurno navesti mnoge njegove čitaoce da dalje sami istražuju ... Tekst nije vezan ni za koji posebni tip kompjutora, već nam netehničkim jezikom objašnjava kako kompjutori rade, kako ih upotrebljavamo i kako možemo upotrijebiti, te što od njih možemo očekivati u bližoj budućnosti, koju kompjutori već sada stvaraju. Štivo je sistematski obogaćeno briljantnim ilustracijama u boji ... Čitalac ga može pratiti redom, od korica do korica, a bez teškoća može i napreskokce birati dijelove koji ga trenutno zanimaju. ..«

Patrick Cain u »ZX Computing« april-maj 1984

»Peter Laurie napravio je izvanredno zrelu knjigu o jednom području koje je tek nedavno izišlo iz dječjih povoja i počelo sticati zrelost. ... Visoke standarde ovog djela upotpunjuju brojni shematski prikazi i ilustracije u boji, koje se odlično uklapaju u tekst ... Knjiga je koncipirana tako da uz malo upornosti možete prokrčiti put od korica do korica, ili — alternativno — možete je koristiti kao priručnik jer su pojedine tema čvrsto zaokružene unutar svojih poglavlja ... Potpuni novajlija u svijetu kompjutora bit će nakon čitanja ove knjige bogatiji informacijama i upućeniji nego li nakon čitanja bilo koje druge knjige u toj vrsti ... «

Linnet Evans u »Practical Computing« februar 1984





Prva knjiga
koja vas jasnim netehničkim rječnikom,
brojnim shematskim prikazima i sjajnim ilustracijame
uvodi u čudesni svijet kompjutora.

Pouzdani vodič i priručnik
od prvih koraka kad se odlučujete nabaviti kućni kompjutor
do zapanjujućih novih granica u grafičkom dizajnu,
robotici i umjetnoj inteligenciji.

Stručne ocjena o engleskom izdanju ove knjige:
»Potpuni novajlija u svijetu kompjutora
bit će nakon čitanje ove knjige bogatiji informacijama
i upućeniji nego li nakon čitanja bilo koje druge knjige
u toj vrsti...«